8558392

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

January 06, 2025

THIS IS TO CERTIFY THAT ANNEXED IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF THE FILE WRAPPER AND CONTENTS OF:

APPLICATION NUMBER: ***13/531,294***
FILING DATE: ***June 22, 2012***
PATENT NUMBER: ***8861349***
ISSUE DATE: ***October 14, 2014***

Certified by

David J. Kappos

Acting Under Secretary of Commerce
for Intellectual Property
and Acting Director of the United States
Patent and Trademark Office

PTO/SB/81 (01-09)
Approved for use through 11/30/2011. OMB 0651-0035
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY OR REVOCATION OF POWER OF ATTORNEY WITH A NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS

Application Number	
Filing Date	October 19, 2008
First Named Inventor	Eric Delangis
Title	A Broadband Comm. Device
Art Unit	
Examiner Name	
Attorney Docket Number	ED 1020

I hereby revoke all previous powers of attorney given in the above-identified application.

☐ A Power of Attorney is submitted herewith.

OR

☒ I hereby appoint Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith:

64828

OR

☐ I hereby appoint Practitioner(s) named below as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith:

Practitioner(s) Name	Registration Number

Please recognize or change the correspondence address for the above-identified application to:

☐ The address associated with the above-mentioned Customer Number.

OR

☒ The address associated with Customer Number: 64828

OR

☐ Firm or Individual Name	
Address	
City / State / Zip	
Country	
Telephone / Email	

I am the:

☒ Applicant/Inventor.

OR

☐ Assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) (Form PTO/SB/96) submitted herewith or filed on _____

SIGNATURE of Applicant or Assignee of Record

Signature		Date	4/19/2009
Name	ERIC M. DELANGIS	Telephone	
Title and Company			

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☐ *Total of ____ forms are submitted.

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RED HAT 1012
RED HAT V. CAS

Electronic Patent Application Fee Transmittal	
Application Number:	
Filing Date:	
Title of Invention:	A BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	ERIC M. DELANGIS
Filer:	Harold C. McGurk
Attorney Docket Number:	ED 1040
Filed as Small Entity	

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Utility filing Fee (Electronic filing)	4011	1	95	95
Utility Search Fee	2111	1	310	310
Utility Examination Fee	2311	1	125	125
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

RED HAT 1012
RED HAT V. CAS

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**530**

RED HAT 1012
RED HAT V. CAS

Electronic Acknowledgement Receipt	
EFS ID:	13088357
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	A BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	ERIC M. DELANGIS
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	22-JUN-2012
Filing Date:	
Time Stamp:	18:43:43
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$530
RAM confirmation Number	6069
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)

RED HAT 1012
RED HAT V. CAS

1	Application Data Sheet	ED_1040_ADS_SB14.pdf	964862 994b809e8ab79f008ced20a4786029a6ad91515d	no	4
Warnings:					
Information:					
2	Specification	ED_1040_Specification.pdf	127710 9d78162e00499dbb19104e104893367a609fa47d	no	24
Warnings:					
Information:					
3	Claims	ED_1040_Claims.pdf	17096 62a2b380d046f41dae2b531506aefb6c6199137a	no	2
Warnings:					
Information:					
4	Abstract	ED_1040_Abstract.pdf	20260 542efa6be03a29d2c0933feaba3be7f1152889f9	no	1
Warnings:					
Information:					
5	Drawings-only black and white line drawings	ED_1040_Drawings_1-9.PDF	2620724 5f2c2bc1b63f197728e97df396e6f7ff589f976b	no	9
Warnings:					
Information:					
6	Oath or Declaration filed	ED_1040_Declaration_Signed.pdf	398549 d945819a4f69fab8f66765c8ae10280935f804a4	no	1
Warnings:					
Information:					
7	Power of Attorney	ED_1040_POA_Signed.pdf	339143 cf2ef6fce40d38f7356ef8acb6dc546139f96174	no	1
Warnings:					
Information:					
8	Fee Worksheet (SB06)	fee-info.pdf	32726 a6e20e5597cf3feb69a24e5174c8dd7762b1e804	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	4521070		

RED HAT 1012
RED HAT V. CAS

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ED 1040
	Application Number	
Title of Invention	A BROADBAND COMMUNICATIONS DEVICE	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Applicant Information:

Applicant 1 Remove

Applicant Authority ◉ Inventor ○ Legal Representative under 35 U.S.C. 117 ○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Eric	M.	Delangis	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Huntsville	State/Province	AL	Country of Residence [i]	US

Citizenship under 37 CFR 1.41(b) [i] US

Mailing Address of Applicant:

Address 1	231 Brevard Blvd		
Address 2			
City	Huntsville	State/Province	AL
Postal Code	35811	Country [i]	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	64828		
Email Address	claymcgurk@cox.net	Add Email	Remove Email

Application Information:

Title of the Invention	A BROADBAND COMMUNICATIONS DEVICE		
Attorney Docket Number	ED 1040	Small Entity Status Claimed	☒
Application Type	Nonprovisional		
Subject Matter			
Suggested Class (if any)		Sub Class (if any)	
Suggested Technology Center (if any)			
Total Number of Drawing Sheets (if any)	9	Suggested Figure for Publication (if any)	6

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ED 1040
	Application Number	
Title of Invention	A BROADBAND COMMUNICATIONS DEVICE	

Publication Information:

☐	Request Early Publication (Fee required at time of Request 37 CFR 1.219)
☐	**Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Enter either Customer Number or complete the Representative Name section below. If both sections are completed the Customer Number will be used for the Representative Information during processing.

Please Select One:	◉ Customer Number	○ US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	64828		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78(a)(2) or CFR 1.78(a)(4), and need not otherwise be made part of the specification.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation of	12581852	2009-10-19

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
12581852	Continuation of	10686375	2005-10-14	7606156	2009-10-20

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
10686375	non provisional of	60418521	2002-10-15

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

Foreign Priority Information:

This section allows for the applicant to claim benefit of foreign priority and to identify any prior foreign application for which priority is not claimed. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(a).

PTO/SB/14 (07-07)
Approved for use through 06/30/2010. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ED 1040
	Application Number	
Title of Invention	A BROADBAND COMMUNICATIONS DEVICE	

Remove

Application Number	Country [i]	Parent Filing Date (YYYY-MM-DD)	Priority Claimed
			(•) Yes ( ) No

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Assignee Information:

Providing this information in the application data sheet does not substitute for compliance with any requirement of part 3 of Title 37 of the CFR to have an assignment recorded in the Office.

Assignee 1 Remove

If the Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information:

Address 1			
Address 2			
City		**State/Province**	
Country [i]		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee Data may be generated within this form by selecting the **Add** button. Add

Signature:

A signature of the applicant or representative is required in accordance with 37 CFR 1.33 and 10.18. Please see 37 CFR 1.4(d) for the form of the signature.

Signature	/H C MCGURK, REG.NO. 34964/			Date (YYYY-MM-DD)	2012-06-22
First Name	HAROLD	Last Name	MCGURK	Registration Number	34964

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

A BROADBAND COMMUNICATIONS DEVICE

RELATED APPLICATIONS

[0001] This application is a continuation of U.S. Patent Application Serial No. 12/581,852 filed October 19, 2009, which is a continuation of U.S. Patent Application Serial No. 10/686,375 filed October 14, 2005, which claims priority to U.S. Provisional Application Serial No. 60/418,521, filed October 15, 2002, all which are incorporated by reference herein.

BACKGROUND OF INVENTION

[0002] The primary function of the RCG is to provide a platform for Competitive Local Exchange Companies (CLECs) to compete with the Local Exchange Companies (LECs) for residential voice and data services. The LECs have a strangle hold on the "last mile" of copper over which almost all residential customers receive their communications services. Today, most CLECs resell the existing services that are offered by the LECs. CLECs make their money by purchasing these services from the LECs at a discount and marking them up to the customer. Since these services are regulated and controlled by the Public Utilities Commission (PUC) the pricing for these services to the CLECs is set. The monthly CLEC discount is approximately $5 to $6 per line per month for basic rate services. This means, in practicality, that a CLEC can only charge an additional $5 or $6 to the residential customer for basic rate services otherwise they would be more expensive than the LEC, and no one would buy the services from them; telephone customers would simply stay with the LEC.

[0003] To be competitive then, CLECs try to sell additional services such as call waiting, caller ID, etc. where they have as much as a 50% margin. Some CLECs also bundle long distance services with their packages as additional incentive to get residential customers to buy local telephone service from them. Since long distance fees are as low as pennies per minute to the customer, there is little profit in long distance for the CLECs. As it is, the pricing structure is such that the LECs will stay in control of the residential market, and the CLECs are nothing more than a marketing arm of the LEC. This structure also keeps the CLECs at an equal competitive advantage, or disadvantage, with respect to each other. The only thing that one CLEC could do to gain more customers is to be a better marketier. Since the CLECs are unable to discount their services because their margins are too slim as it is, the net effect is that the

RED HAT 1012
RED HAT V. CAS

consumer looses by not really getting competition in the residential markets, and prices are kept artificially high.

[0004] For the CLECs to be truly competitive in the local telecommunications markets, they must be able to do more than simply resell the LECs services. CLECs must be able to offer many more features or services than those that the LEC provides, and they must be able to do this at a lower cost, in order to attract new customers.

[0005] So, what new services can the CLECs offer? As things stand today there are none, with the exception of DSL or Cable services which require tremendous infrastructure costs on behalf of the CLEC. However, by deploying an RCG in a residence, CLECs can offer advanced new calling features and services over the existing POTS line. Their costs for the POTS line from the LEC remains the same, but they can now charge for high bandwidth data services as well as additional calling and messaging features that the LECs cannot offer. This is their competitive advantage.

Competition:

[0006] Many companies are currently trying to provide VoIP services to residential customers over high-speed or broadband backbones such as DSL and cable modems, over Internet connections via a computer. The high-speed services require expensive infrastructure enhancements to the local telephone companies' facilities and cable plants, as well as expensive modems at the customer's end. These services are also difficult to deploy and too complicated for the average user. Internet based VoIP services suffer from Internet delays caused by a lack of true Quality of Service (QOS) capabilities across the Internet. Existing VoIP services also require access to a computer and a technical knowledge to be able to make it work. Because of these problems, VoIP services are primarily targeted at the corporate market place where private IP networks that can carry VoIP services with a true guaranteed QOS exist, or can be controlled.

[0007] The only available alternatives for the residential user for high speed services are DSL and cable modems. DSL and Cable based communications reach only a small segment of the population, and growth has failed to live up to initial expectations. The recent collapse of the internet boom has led to the failure of many DSL service providers and has effectively eliminated DSL as a potential service for most POTS users for the immediate future.

SUMMARY OF THE INVENTION

[0008] The RCG combines the capabilities and features of IP routers, Class 5 circuit switches and wireless LANs in a small stand-alone external modem like device. Additional capabilities for a hands-free speakerphone device with a large touch display are also detailed. The RCG can employ a large display with sufficient resolution to provide a functional user interface to access all of the advanced features offered by the RCG, but also to be able to provide video telephone services.

[0009] The device is placed in a residence and connected to the primary POTS line coming from the LEC. Up to three standard telephones or FAX machines (or any combination) are typically connected into the RCG. It also has a Universal Serial Bus (USB) and/or a 10-100 Ethernet interface for connection to a computer or a Local Area Network (LAN). In addition, it has an 802.11b/g wireless interface for wireless home networking as well as broadband-to-the-home services.

[0010] The system is installed in a customer's home by simply plugging it into the existing incoming telephone line that provides POTS services from the LEC. One or more standard telephones are then plugged into its three line-output jacks. This is the simplest and most basic usage. In this implementation, the user only is interested in gaining additional telephone lines into the household. No computer connection is required in order to use the system. However, if data communications are desired, then there are three additional interfaces for computer connectivity. Typically either an Ethernet port, USB port or firewire (IEEE 1394) port is used to connect to a PC or LAN. The 802.11b/g interface is for wireless communications between the RCG and PCs, other RCGs as well as any other types of compatible wireless devices. A version of the RCG includes a speakerphone and video camera to provided for video telephone services.

Services provided by the RCG:

[0011] The following discussion refers to figure 1. The RCG has a main system processor that is responsible for all house keeping and traffic routing tasks. The Main CPU 19 routes data to and from the Ethernet Port 11, USB Port 12, Modem/DAA 41, DSP Engine 33, 802.11b/g Module 36 and the Display and Keypad 10.

[0012] The RCG provides a total telecommunications communications services package for residential and small business customers. With the RCG, a CLEC can offer its customers dialtone for up to three separate telephone numbers per residence (the limit of three telephone numbers is an arbitrary number based upon typical residential usage and is not a limit imposed by the system), all local and long distance services, 24/7 always-on Internet access, a host of custom messaging and calling features, as well as Broadband access and lifeline service. All of these services can be offered over a customer's existing POTS line, and it doesn't require any network infrastructure build out like DSL and Cable services. Because there is no expensive network equipment required, it can be readily deployed throughout the country to expand the CLECs customer base quickly.

[0013] The RCG is designed to enhance, not replace, the existing telephone line, enabling it to provide new combined packetized voice and data services. Meanwhile, all existing telecommunications infrastructure and equipment will continue to work seamlessly with the RCG. Aside from simply installing the low cost RCG device in the home telephone wiring, no other equipment needs to be purchased by either the home owner or telephone companies. This eliminates the tremendously expensive infrastructure costs for communications companies that are trying to provide voice services using DSL or cable modems.

The RCG VIP Home Gateway features:

[0014] Up to three telephone lines i.e. numbers, over a single POTS line and/or wireless connection.

[0015] Ethernet, USB and IEEE 1394 computer interfaces.

[0016] Full G1, G2 and G3 FAX support.

[0017] Full support for all CLASS services.

[0018] Broadband networking for home computers and appliances as well as wide area networking using a combination of a plurality POTS lines connected to a plurality of RCGs and/or a direct wireless connection to a wireless access point. (These interfaces allow smart devices to communicate to other devices within the home, or over the Internet through the RCG.)

[0019] Automated, remote upgradability and configuration. (Requires the customer to do nothing.)

[0020] Transparent user operation. (User doesn't have to know that any new technology is being employed)

[0021] 100% compatible with existing public telephone system.

[0022] Video Telephone capabilities.

[0023] No local or long distance usage charges.

[0024] No Internet access fees.

[0025] Lower monthly communication costs.

[0026] Multiple telephone calls at the same time.

[0027] Voice messaging, follow me features, call forwarding, automatic call back, conferencing, custom call routing. Home networking capabilities, all CLASS calling features, high speed broadband networking, continuous 24/7 internet access. and many others.

[0028] All of these features are available even while connected to the Internet.

[0029] Very low cost of entry into the market, especially as compared to broadband alternatives.

[0030] New calling features that are not currently available to home users.

BRIEF DESCRIPTION OF THE DRAWINGS

[0031] Figure 1 shows the RCG functional block diagram with the preferred embodiment of an 802.11 wireless connection.

[0032] Figure 2 shows the network topology for native RCG to RCG communications.

[0033] Figure 3 shows the network topology for an RCG to PSTN call.

[0034] Figure 4 shows the RCG top view with associated functions.

[0035] Figure 5 shows the RCG rear view and associated connections.

[0036] Figure 6 shows a typical application with peripheral equipment connections.

[0037] Figure 7 shows an ad-hoc wireless LAN configuration with ppp multilink data distribution and wireless connectivity.

[0038] Figure 8 shows an ad-hoc wireless LAN configuration with ppp multilink data distribution and wireless connectivity with connection to a broadband neighborhood access point (Wi-Fi hot spot).

[0039] Figure 9 shows an ad-hoc wireless LAN configuration with ppp multilink data distribution and wireless connectivity with wireless data hopping with POTS connections through numerous different Class 5 offices.

DETAILED DESCRIPTION OF THE INVENTION

[0040] The preferred embodiment is detailed herein using the ubiquitous 802.11 wireless protocol. It is by no means insinuated that this is the only wireless scheme that can be used by this device; rather, it is intended that any wireless transmission scheme can be used for wireless communications between this device and other wireless devices as various applications may dictate.

System Initialization

[0041] Refer to figures 1, 2 and 3 for the following discussion. When the RCG is first plugged into an AC power outlet the Main CPU 19, performs a Power-on Self Test (POST) routine. Once the POST completes and the RCG is connected to an active POTS line, connected to the LEC, by means of the Incoming POTS Port 40, the RCG automatically initiates a modem connection via the Modem/DAA 41, to a registration service via a toll free 800 telephone number. The registration service connects directly to the service provider's sales or customer service offices via a VoIP connection. An operator or an automated service answers the call and sets up a new account for the customer. Once the account is set up, the service provider's network automatically sends control information to the device to store the required routing and feature/service information in its non-volatile memory of its Main CPU 19. As part of this information, a local dialup telephone number is sent to the device that it will use for local network access in the same fashion as a standard ISP connection is made from a home computer. The URL of the Softswitch/SIP Proxy Server and Registrar 51 that will be controlling this particular device is also stored at this time. During the service activation and configuration process, the customer service operator can send test requests to the device, as well as call into it on another telephone number (assuming an additional telephone number is part of the users class of service) or the speakerphone to verify that it is configured and working properly. The final step of the configuration and initialization process is to register the device with the appropriate

RED HAT 1012
RED HAT V. CAS

Softswitch/SIP Proxy Server and Registrar 51 so that the Softswitch/SIP Proxy Server and Registrar 51 knows that it is the controlling SIP Proxy for this RCG. Also configured at this time is the username/password verification server (this is a standard network device and is not shown) that is used to allow the RCG's modem to make a connection onto the service provider's network. Once the RCG configuration is completed, it disconnects the call by hanging up the Modem/DAA 41, then dials the new local network access number that had been sent during the configuration process and establishes a constant local network connection. This connection is maintained continuously on a 24/7 basis, and is immediately reestablished whenever it drops. Should the RCG become unable to reestablish this connection after several tries, it will switch over to failsafe mode and inform the user via the Display and Keypad 10. The RCG will try to reestablish the connection every few minutes until it finally succeeds.

[0042] By default, the first telephone port, POTS 1 Port 30, is assigned as the termination port for the existing POTS service that is connected to Incoming POTS Port 40. This is the existing telephone service that the user already has as his primary telephone service from his LEC. POTS 1 Port 30, typically has connected to it a standard telephone set that does not require any power (other than that derived over the POTS interface) to operate properly. POTS 1 Port 30 is the Lifeline telephone port for the user. In the case that all power is lost to the RCG, POTS 1 Port 30 will fail to a direct connection to the Incoming POTS Port 40 telephone line that comes into the RCG from the LEC to provide uninterrupted telephone services to the user. In this state, however, none of the enhanced VoIP services of the RCG are available to the user since the user simply has a stand POTS telephone service to the LEC.

[0043] Two additional and unique telephone numbers can be assigned to the other two telephone ports, POTS 2 Port 29 and POTS 3 Port 28, by the service provider at any time. These numbers are drawn from a pool of available telephone numbers assigned to the service provider. POTS 2 Port 29 and POTS 3 Port 28 must to have standard telephone numbers assigned to them so that the PSTN is able to route calls from its network to these ports. For calls originating from IP based telephones, these numbers are not necessarily needed since POTS 1 Port 30, POTS 2 Port 29 and POTS 3 Port 28 are uniquely identifiable to IP networks via their IP addresses assigned through the Dynamic Host Configuration Protocol (DHCP) at the time that the RCG establishes a connection to the service provider's network. Since POTS 2 Port 29 and POTS 3 Port 28 must have a telephone number assigned to them, it is a matter of convenience to use the

assigned telephone numbers as part of their SIP address for IP routing of their data. In this way, all three user POTS ports, POTS 1 Port 30, POTS 2 Port 29 and POTS 3 Port 28 are all addressed the same manor.

[0044] At this point, the user is able to take advantage of all of the RCG's features that have been activated according to his service agreement. The user can at any time connect additional telephones, a computer or a computer network to the RCG. The user will only be able to use those telephone ports (POTS 1 Port 30, POTS 2 Port 29 or POTS 3 Port 28) that have been activated, and assigned unique telephone numbers by the service provider. Any type of standard telephone can be connected to the three POTS ports, POTS 1 Port 30, POTS 2 Port 29 and POTS 3 Port 28. In the event that a cordless telephone or a telephone that derives its power from an AC outlet is connected to POTS 1 Port 30, the lifeline functionality of this port will be ineffective since the telephone itself will not operate once power is lost.

[0045] The speaker phone feature of the RCG, comprised of Speakerphone Drivers 22, Microphone 24, Speaker 25 and DSP Engine 33, can be used on any of the activated telephone numbers and is by default enabled to operate on POTS 1 Port 30. Once either of the other two telephone ports, POTS 2 Port 29 or POTS 3 Port 28, is activated, the speakerphone function is activated for those lines as well. To use the speakerphone for any assigned telephone number, the user simply presses either the Line Select Button 79-81 for Line 1, Line2 or Line 3 on the face of the RCG, and the speakerphone function is activated on that line. This is convenient for answering any call directly at the RCG. The RCG may or may not have a number keypad for originating calls. This feature may be added or removed based upon the target market price points.

[0046] The determination as to which Inter Exchange Carrier (IXC) the RCG will route its data through is determined by any of a number of criteria including, but not limited to, whether the RCG is a private labeled product, or which carrier has the lowest cost routing, or if a specific carrier is specified by the customer, or and number of other considerations. The RCG can be programmed to change its routing path to any IXC at any time by the service provider. Complicated routing algorithms can be programmed into the RCG to control which IXC it will use at any time.

Failsafe Mode

[0047] During any time that the RCG is in failsafe mode i.e. Lifeline failsafe logic 39 is activated, the RCG will only be able to receive or initiate telephone calls via the telephone connected to its primary telephone port, POTS 1 Port 30 or through its speakerphone feature. In this condition, all activated telephone numbers are ported via the PSTN to the primary (first) telephone number for delivery to the user's telephone that is connected to POTS 1 Port 30. While in failsafe mode, the modem is not activated, and the RCG appears to the local Class 5 office as a standard telephone, and the PSTN will route all incoming calls to the RCG as it normally would to any standard telephone.

Call Flow 1:

[0048] Dialed call origination from a telephone connected to an RCG POTS port and terminating at another user's RCG POTS port:

[0049] When a user picks up a telephone connected to any of the three POTS ports, POTS 1 Port 30, POTS 2 Port 29 and POTS 3 Port 28, an off hook condition is detected by the SLIC/CODEC Interface 27 that is connected to the DSP Engine 33 by way of the DSP Audio Interface 31. The DSP Engine 33 responds by sending digitally generated dialtone to the appropriate POTS interface over the DSP Audio Interface 31. (Note: all BORSCHT telephony functions are performed by the DSP Engine 33 and the SLIC/CODEC Interface 27.) Upon hearing the dialtone, the user enters the desired telephone number on his telephone keypad. The Dual Tone Multi-Frequency (DTMF) digits that the user dials are decoded within the DSP Engine 33. The dialed digits are transferred to the Main CPU 19 via the CPU/DSP Communications Path 34. The Main CPU 19 polls the DSP Engine 33 approximately every two milliseconds looking for data. The Main CPU 19 polling rate varies according to the process that is being performed. The CPU/DSP Communications path 34 also can generate an interrupt to force data transfer between it and the Main CPU 19.

[0050] The Main CPU 19 generates a SIP Invite IP packet using the dialed digits as the called destination address and the URL of its local controlling SIP Proxy Server which was programmed into the RCG during the configuration process. The Main CPU 19 passes this SIP packet off to the Modem/DAA 41 for transmission over the network. The RCG Modem/DAA 41 is connected to and communicating with a modem in a modem bank located in the Inter

Exchange Carrier's (IXCs) facility. This facility may actually be located in the LEC's physical plant or it may be physically located at another location.

[0051] Referring to Figure 2, User A RCG 44 is connected via its modem to the LEC's local Class 5 Office A 47. In this example, the service provider's connections to the IXC A Modem Bank & Router 48 are through IXC A Inter-Office Trunks 46. The service provider may or may not be the IXC. If the service provider does not have its own network, then it will purchase capacity from an IXC and route its traffic to that IXC's facilities. The SIP Invite IP Packet that is sent from User A RCG 44 transits the LEC's Class 5 Office A 47 facilities and is routed to the IXC A Modem Bank/Router 48 via the IXC A Inter-Office Trunks 46. It is the job of the router to take the IP packet and route it on to its intended destination, in this case the appropriate SIP proxy server 51. The router uses the URL for the Softswitch/SIP Proxy Server and Registrar 51 contained in the SIP Invite to know where to route the packet. (Note: The Softswitch, SIP Proxy Server and Registrar are logical devices that may reside in a single computer system, or may be implemented as completely separate devices. In this example, the Softswitch, SIP Proxy Server and Registrar are all shown as a single physical device running these three separate processes.) The IXC A Modem Bank/Router 48 sends the SIP Invite IP packet on to Softswitch/SIP Proxy Server and Registrar 51 via an IP based network route 52.

[0052] The SIP Proxy Server process running on Softswitch/SIP Proxy Server and Registrar 51 performs a look up to determine the termination destination for the call using the dialed digits passed along in the SIP Invite IP packet. In this example, the terminating device for User B RCG 56 is via IXC B's Modem Bank/Router 49 then through Class 5 Office B 50 and over User B POTS line 55 and finally to User B RCG 56. Not covered in detail here is the SIP message handling since it is completely covered in IETF RFC2543. This example shows that both the originating and terminating ends of this call are controlled by the same Softswitch/SIP Proxy Server and Registrar 51 device. Where the originating and terminating ends of a call are controlled by different Softswitch/SIP Proxy Server and Registrar devices, an additional level of SIP messaging is used between the two SIP Proxy Server processes of each of the Softswitch/SIP Proxy Server and Registrar devices to locate and control the terminating location of the call. Again, that information is covered in IETF RFC2543 and is repeated here.

[0053] After the call routing is setup using SIP, and User B answers the call, the Softswitch/SIP Proxy Server and Registrar 51 device sends IP address of the originating User A

RCG 44 to the User B RCG 56 and the IP address of User B RCG 56 to User A RCG 44. Once both RCG's have the other's IP address, they then communicate directly to each other using Real Time Protocol (RTP), and the Softswitch/SIP Proxy Server and Registrar 51 device is no longer part of the connection. This direct communication is shown as IP based RTP Stream 57 between IXC A Modem Bank/Router 48 and IXC B Modem Bank/Router 49.

[0054] On the receiving end of the call, referring back to Figure 2, User B RCG 56 receives the SIP Invite data stream through its Modem/DAA 41 and passes it on to the Main CPU 19. The Main CPU 19 reassembles the SIP IP packet and interrogates the header information to determine which POTS port, POTS 1 Port 30, POTS 2 Port 29 and POTS 3 Port 28 the call is destined for based upon the dialed number contained in the SIP Invite message. The Main CPU 19 commands the DSP Engine 33 to control the SLIC/CODEC Interface 27 to apply ringing voltage to the selected POTS port to ring the phone. A corresponding SIP 180 (ringing) message is sent back to User A RCG 44 in the reverse direction. When the phone attached to the selected POTS port is answered, the off hook condition detected by the SLIC/CODEC Interface 27 is sent through the DSP Engine 33 to the Main CPU 19 that sends the SIP 200 OK message back to User A RCG 44 to let the RCG know that the call was answered by User B RCG 56 and to turn off the audible ringing tone for User B. Once the call is established, User A RCG 44 sends and ACK F12 SIP message to User B RCG 56 indicating that User A RCG 44 understands that the call setup has been completed.

[0055] The two RCGs now exchange SIP messages that include the type of voice encoding that will be used such as G.729, G.723.1, G.711 etc to configure their RTP streams so that they can reliably transfer the user information between them. At this point the call (voice data) is sent between both RCGs and a normal telephone conversation takes place between the user A and user B.

[0056] Call teardown is accomplished by either user hanging up their telephone. The RCG of the terminating end (the end that hangs up the phone first) detects the on-hook condition and sends a SIP Bye message to the other end. The other end upon receiving the SIP Bye message responds with an OK 200 SIP message and disables the appropriate POTS port that the user was using.

[0057] The Speakerphone interface can be used for either end of a telephone conversation in exactly the same way as described above except that the corresponding POTS port that the call was intended for is disabled if the call is answered via the speakerphone.

Call Flow 2:

[0058] Calls between an RCG device and a PSTN telephone:

[0059] Refer to figure 3 for the following discussion.

[0060] In the previous call example, the PSTN network had nothing to do since the call was over a previously established modem connection between two RCGs. In the cases where a call originates or terminates to a standard telephone serviced by a Class 5 office, the process is a bit more complicated.

[0061] When a call originates from a user via an RCG, such as User A RCG 44, the call handling practices are exactly the same up to the point that the SIP Proxy Server process in the Softswitch, SIP Proxy Server and Registrar 51 performs a lookup to determine how to route the call. In this case, the SIP Proxy Server process determines that the call terminates at a standard PSTN telephone attached to the Class 5 Office B 50. Since the call originates as a native VoIP service, it must be converted to a standard switched circuit based service to be able to be delivered to User B's telephone 61. The conversion is performed by the IXC's Media Gateway 60 device. In this example, the SIP Proxy Server process routes the call via the IXC's Media Gateway 60 device instead of through the IXC B Modem Bank/Router 49 as before. The SIP Proxy Server process sends a SIP Invite to the IXC's Media Gateway 60 device, as well as other control information, to direct the Media Gateway 60 device to terminate the call and to apply the appropriate voice compression algorithm to the media stream to convert it between switched circuit PCM format and the selected VoIP protocol. At the same time, the Softswitch process of the Softswitch, SIP Proxy Server and Registrar 51 device sends commands to the Media Gateway 60 device to cause it to route the call's PCM data to a specific channel on the IXC B Interoffice Trunk 54. The Softswitch process then generates an SS7 ISUP message that is sent over the PSTN SS7 signaling network to the Class 5 Office B informing it of the incoming call and on which circuit the call has been placed. It is up to the PSTN at this point to ring the User B telephone 61 and process the call between the IXC's Media Gateway and the User B Telephone 61. The Softswitch process handles the signaling conversion between the PSTN's

SS7 network and the IXC's SIP messaging, as well as any other proprietary signaling schemes utilized by the IXC to control its network. The standard SS7 call control messages, such as ringing, busy, ring back etc., all have a comparable SIP call flow control messages. The Softswitch simply maps the SS7 messages into SIP messages and visa-versa while routing the messages to the appropriate network equipment.

[0062] As in the case of a call between two RCGs, the RTP media stream (the actual call data) is routed directly be the IXC A Modem Bank/Router 48 and the IXC's Media Gateway 60 via an IP based RTP Stream 57.

[0063] Again, call teardown is accomplished by either user hanging up their telephone. Depending which end hangs up the phone first, the associated network will generate the appropriate messages which will eventually make it back to the Softswitch process. The Softswitch process will relay the appropriate call control messages between the PSTN and the IXC's SIP based control system.

[0064] A telephone call originating from a PSTN telephone is configured and setup in the same fashion as the above description except that the originating messaging begins within the PSTN network instead of the IXC's network.

Personal Computer based communications:

[0065] Where the RCG is connected to a Personal Computer (PC) through either its USB Port 11 or Ethernet Port 12, the RCG operates exactly as a cable or DSL type modem device. The RCG is simply a pass through for the computer's data. The RCG receives the IP packets (of whatever type) from the computer and sends the information unchanged to the Modem/DAA 41 interface for transmission over the Internet. Received information from the Internet is processed in the reverse way.

Dynamic Bandwidth Allocation:

[0066] When a modem connection is established between the RCG and the "Network", the total available bandwidth of this connection is used in a shared/prioritized fashion. Voice services (calls) take precedence over data services. Therefore, when no calls are in progress all of the available bandwidth is used by data type services such as surfing the Internet or uploading/downloading files. However, when a call is to be passed through the RCG, the

required bandwidth for the call is deallocated from the data services and prioritized for the telephone call. This necessary because the Quality Of Service (QOS) is important for effective communications over electronic networks. Since the voice data is carried over a "data centric" network topology (an IP based network), call quality can suffer greatly. Every effort must be taken to eliminate or reduce the packet delay through every network element to be able to provide the best call quality. For data applications such as surfing the Internet or downloading files packet delay is acceptable and expected, and it does not affect the "quality" of the data.

[0067] Since all voice and data is transmitted as packets, there are no "time slots" or other type of "channelization" associated with the transmission of this data over either the POTS line or the wireless interface. Channelization of the voice data does occur within the SLIC/Codec Interface – 27 but is converted into IP packets by the DSP engine – 33 and is handled as packets from that point on.

[0068] Regardless of whether the call is originating or terminating at the RCG, the bandwidth allocation process is the same. When the SIP call setup messages are processed, the type of voice compression that will be used is defined. The RCG Main CPU 19 handles the IP packet flow between the DSP Engine 33, USB Port 12, Ethernet Port 11, 802.11b/g Module 36 and the Modem/DAA 41 devices giving priority to the voice packets to and from the DSP Engine 33 and the Modem/DAA 41 interfaces. Voice compression algorithms create IP packets at a specified number of packets per second. These packets are handled by a timed process in the RCG. This process is started during call setup so that the packets are handled at the correct rate based upon the compression algorithm's packet generation rate. There is no priority assigned between the voice packets of each of the three different POTS Ports 28-30. When more than one voice call is in progress, they are handled in a round robin fashion based upon the packet rate for the compression algorithm used. If, for example, two calls are using a compression scheme that generates packets every 2 milliseconds, and the third generates packets every 4 milliseconds, then the calls utilizing the faster packet rate will be polled at twice the rate as the call using a slower packet rate but they are not given priority "over" the other call. Only when there are no voice packets ready for transmission to the Modem/DAA 41, i.e. during the approximately 2 to 4 millisecond packetization period between packets, will any pending data packets received from the USB Port 12, Ethernet Port 11 or the 802.11b/g Module 36 be sent to the Modem/DAA 41 for transmission over the network.

[0069] Thus, the POTS bandwidth is allocated on a per packet basis, not on a "time slot" or "channelization" or even a "frequency" basis as it the case with T-Carrier, DSL or channelized DSL systems. The same is true for the wireless interface, whether it is carrying pure data or voice.

[0070] Security is currently a big concern in today's networks, and especially with wireless 802.11 transmissions. For this reason, the RCG defaults to route voice data over the users POTS connection via Incoming POTS Port 40. As security concerns regarding wireless communications transmissions are addressed and alleviated, voice call data can, and will, be routed over the wireless interface via the 802.11 b/g Module 36.

FAX Transmission:

[0071] Group I, 2 and 3 fax transmission is completely supported by the RCG. Fax transmission has the same priority as other voice communications. That is to say that FAX transmission is treated the same as a voice call and takes precedence over native IP data traffic.

Broadband Over POTS:

[0072] One of the primary functions of the RCG is to be able to transfer large files such as video files over the POTS lines to a residence. The ability to do this is limited by the maximum of 56Kbps due to the design of the digital Class 5 office linecards and network transmission schemes based upon T carriers. Typical Class 5 linecard design samples the POTS lines at a rate of 8,000 samples per second. Typically, each sample is digitized into 12 to 16 bits by a codec device. The digitized sample is then companded to an 8 bit sample (1 byte) which makes up the PCM data stream that is transferred over the PSTN. Therefore, each POTS line is sampled at an effective rate of 64Kbps. Since legacy T1 transmission equipment uses robbed-bit signaling, the maximum guaranteed transmission rate for the PSTN as a whole is 56Kbps. Because of this limitation, standard POTS lines can only guarantee a maximum of 56Kbps transfer rate under the best conditions. DSL equipment gets around this problem by installing DSL Access Multiplexing (DSLAM) equipment at the Class 5 office end of the POTS line. By routing around the linecard, as well as routing the data directly over the Internet or other private IP based networks, and therefore the 64Kbps sampling rate, DSL is not bandwidth limited like PSTN traffic is. DSL, however, suffers from the very high costs associated with the purchase,

installation, maintenance, floor lease costs, etc. DSL also suffers from very high deployment costs resulting from the labor intensive process to turn up each line. Each DSL "POTS" line must be inspected, conditioned and all load coils removed.

[0073] The RCG gets around the 56Kbps POTS limitation as well as the DSL problems by using the standard POTS lines as they are, and not requiring any additional equipment to be installed at the Class 5 end of the POTS line. This technique simplifies deployment and drastically cuts costs.

Multilink PPP connections:

[0074] Referring to figure 7, the RCG implements broadband over POTS by employing an 802.11 b/g wireless interface to configure a multilink PPP bundle – 87 (per RFC 1990) comprised of the POTS lines connected to up to 32 RCGs. The limitation of 32 RCGs in a multilink PPP bundle is an arbitrary number and can be scaled up or down as required to support changes in network equipment capabilities. An 802.11b/g interface – 87 is shown as the preferred embodiment, since its 11/54 Mbps interface is considered sufficient bandwidth at this time also, its transmission distances are sufficient for implementation of this device, as opposed to other shorter ranged wireless transmission schemes. However as time goes on, some other wireless interface may be more desirable. There is nothing special or specific to the 802.11 wireless interface that makes it particularly suited to this application. Any wireless interface scheme can be used to implement this device.

[0075] The 802.11b/g interface – 88 on an individual RCG is also used to provide a peer-to-peer wireless home networking environment to other 802.11 compatible wireless component within the residence (such as a hand-held computer device) while acting as an always on dialup Internet gateway over the POTS line – 40 for the other wireless devices. An RCG that is transferring a very large file can request the services of 31 (in this embodiment) other RCGs to assist in the file transfer using their respective POTS lines. A single RCG can therefore gain the combined bandwidth of up to 32 POTS lines transferring data at as much as 56Kbps each to transfer information. This is roughly equivalent to a transfer rate of 1.79Mpbs in the inbound direction (towards the RCG) and approximately 1.075Mbps in the outbound direction (towards the PSTN).

[0076] The multilink PPP connection – 87 is initiated by an RCG when it determines that the file (or streaming service) that it is to download (receive) is of sufficient size or requires a minimum amount of bandwidth to justify the use of a multilink PPP connection. This determination can be made on many different criteria such as the file size, type of data being transmitted, the type of compression if any, a Quality-of-Service requirement or any other pertinent information. (Since a certain amount of overhead is required to set up a Multilink PPP connection, it doesn't make sense to use it for small or low data rate transfers.)

Multilink PPP bandwidth allocation and initialization:

[0077] Referring to figure 7, an RCG – 85 that decides to initiate a multilink PPP session with a remote computer i.e. internet URL, requests a multilink PPP session with that URL resource via Link Control Protocol (LCP) session management. The initiating RCG - 85 and the remote server computer (not shown) will negotiate the pertinent multilink parameters such as compression, packet size, sync/async, link quality monitoring, bit stuffing etc. As part of a successful multilink PPP session setup, the initiating RCG –85 receives a multilink PPP session ID, or "magic number", which is used to identify this particular multilink PPP session and is also used for adding or removing links from the multilink bundle – 87 . Once the multilink bundle is setup, the initiating RCG – 85 begins data transfer across its POTS interface – 90 with the far end computer system. The initiating RCG – 85 then looks up the addresses of up to 31 (in this embodiment) other RCGs in its router tables to use for the multilink PPP transfer. (If a Neighborhood Access Point is available the RCG could request service through it instead of setting up a Multilink PPP connection – see figure 8.) The initiating RCG - 85 determines the number of remote RCGs - 86 that it wants to use to get the desired bandwidth for the transfer and sends multilink PPP service requests via the 802.11 b/g wireless connections – 88 to the selected remote RCGs - 86. The selected remote RCGs - 86 respond back to the initiating RCG with either an acceptance or denial of multilink PPP service request. This decision is made based upon the remote RCG's currently available POTS bandwidth and the current local demands for that bandwidth. The remote RCG – 86 makes a decision to participate in a multilink PPP service for an initiating RCG if its POTS line – 91 has a current average bandwidth utilization of less than some arbitrary limit such as 8Kbps to 16 Kbps. This 8Kbps to 16Kbps bandwidth utilization represents about two concurrent voice conversations using standard voice

compression techniques, or some small amount of data transmission occurring on the remote RCG's POTS interface – 91 or a combination thereof. Assuming that the remote RCG – 86 has a modem connection speed somewhere between 33Kbps to 56Kbps, this would leave approximately 17Kbps to 40Kbps of POTS bandwidth that the remote RCG – 86 could offer to the initiating RCG – 85 for use in its multilink PPP bundle - 87. Of course, if there is no activity on the remote RCG – 86 POTS line, then the entire bandwidth will be used for the multilink transfer. If the remote RCG – 86 accepts the multilink PPP service request from the initiating RCG -85 it also sends its currently available bandwidth on its POTS line - 91 that the initiating RCG –85 will have available to it for the multilink PPP transfers. The initiating RCG then selects which of the responding remote RCGs - 86 it will use for the multilink PPP bundle - 87 and sends them a multilink initialization packet containing the URL of the destination as well as the multilink PPP session ID (magic number) and any other pertinent information required for the remote RCG to join in the existing multilink PPP bundle - 87. Each remote RCG POTS interface – 91 participating in a multilink PPP bundle, represents a logical "link" in the initiating RCG's multilink PPP bundle - 87. The remote RCGs – 86 then send a multilink PPP packet (over their respective POTS lines - 91) to the destination URL with the multilink PPP session ID and request to be added to that multilink PPP bundle - 87. The destination URL accepts the request of each of the remote RCGs - 86 and begins sending data packets to each over their respective POTS lines - 91. When the remote RCGs –86 receive data packets sent over their respective POTS lines – 91, that are destined for the multilink PPP bundle - 87, they simply relay the data packets over their 802.11b/g interface - 88 to the initiating RCG -85. It is the responsibility of the initiating RCG – 85 to perform all housekeeping tasks associated with the multilink transfer such as detection of lost packets and data packet reassembly, if those functions are required. Typically, lost packet detection and retransmission and reassembly are performed at higher OSI layers by the requesting software such as a browser. The remote RCGs – 86 are simply data transport nodes and do not perform any data processing on data received over a multilink PPP bundle destined for another RCG.

[0078] The initiating RCG- 85 of the multilink PPP bundle -87 continually monitors the bandwidth of each remote RCG's -86 multilink link which together comprise its multilink bundle – 87. When the available bandwidth for the multilink PPP link of an individual remote RCG -86 falls below an arbitrary minimum set by the initiating RCG – 85, the initiating RCG – 85

RED HAT 1012
RED HAT V. CAS

instructs the offending remote RCG, via its 802.11b/g wireless interface – 88, to remove itself from the multilink bundle. The remote RCG - 86 will then notify the remote computer (URL) to be removed as a link in the current multilink bundle on which it is participating. Alternatively, a remote RCG - 86 can inform the initiating RCG – 85 of local demands on its POTS bandwidth, and request that the initiating RCG – 85 remove it (the remote RCG – 86) from the participation in the initiating RCG's multilink bundle – 87. The remote RCG – 86 will not unilaterally remove itself from participation in a multilink bundle without prior notification to the Initiating RCG - 85. The remote RCG – 86 must get authorization from the initiating RCG – 85 to remove itself from the bundle, since the initiating RCG – 85 will have no way of knowing if the loss of data from that remote RCG is due to some sort of failure or from the remote RCG simply removing itself from participation in the multilink bundle - 87. Also, for services that require a minimum Quality of Service (QOS) such as streaming video, the initiating RCG – 85 needs to control the multilink bundle – 87 bandwidth, and it can not do that if remote RCGs - 86 drop out of the multilink bundle – 87 at will. When a remote RCG – 86 requests to be removed from participation in a multilink bundle - 87, the initiating RCG – 85 must immediately grant the request. If the initiating RCG – 85 were to not grant the request, then the remote RCG – 86 could, in worst case, delay or drop incoming multilink packets destined for the initiating RCG – 85. If this were to happen, the initiating RCG – 85 will have missing packets and could over run its reassembly buffer (if reassembly is being performed by the RCG) causing catastrophic data loss and potentially have to restart the entire transmission. In the least case, the initiating RCG – 85 would have to initiate retransmission of potentially many packets which it assumes to be lost, but may simply be delayed by the remote RCG – 86. Once a remote RCG – 86 has been removed from participating in a multilink bundle – 87, the initiating RCG – 85 will then look up another remote RCG in its router tables that can be used to participate in the multilink PPP bundle – 87, and request that the new remote RCG -86 to join the existing multilink bundle – 87. This is necessary to maintain a minimum Quality of Service. Since RCGs prioritize their bandwidth requirements based upon their own "local" service requirements, their participation in any multilink PPP service is available only to bandwidth that is not required for its own current local demands. Therefore, a remote RCG's – 86 bandwidth that is available for multilink services can change at any time. The dynamic reassignment of remote RCGs that participate in a multilink bundle is also required to prevent a remote RCG – 86 from receiving packets so slowly

that the originating RCG – 85 interprets this delay as lost packets. When the initiating RCG – 85 determines that a packet has been lost, and it is performing packet reassembly, it will, itself, request that the remote computer retransmit the lost packet. The initiating RCG will not ask any remote RCG – 86 to perform retransmissions or any other kind of data computations or manipulations.

[0079] Refer to figure 8 for the following discussion. In areas where there are existing 802.11 network equipment such as Neighborhood Access Point - 93, or where there is sufficient bandwidth requirements from an installed base of RCGs to warrant the deployment of a Neighborhood Access Point - 93, the RCGs will automatically detect the presence of the compatible neighborhood access points and register with that equipment for transport services. In this case, an RCG has an alternative broadband service route in addition to the multilink option. An RCG can communicate directly with a local 802.11 Network Access Point - 93 that is connected to the PSTN or another carrier's network via some sort of high bandwidth backbone such as a T-Carrier or optical carrier connection (T1, T3, OC3, OC12 etc.). The RCGs can communicate directly with neighborhood access points at their full 11/54 Mbps data rates to provide up to that full bandwidth to the user. This functionality is, of course, subject to the Network Access Point – 93 allowing access to its resources to the RCG according to the 802.11 wireless protocol.

[0080] This architecture is important since it provides an avenue to relieve the local class 5 offices of congestion that can be caused by a wide spread deployment of RCGs within the class 5 service area. Since each RCG requires a constant 24/7 modem connection, this ties up the class 5 switch in a fashion that it was not designed for. This could lead to wholesale class 5 office expansions which would be extremely expensive, and potentially prohibitive for the Local Exchange Carriers (LECs). By deploying neighborhood access points that operate at 54Mbps (802.11a/g), in areas where the RCG concentration starts placing too high of a demand on the local class 5 office, the bandwidth from the RCGs can be aggregated by the neighborhood access points into large capacity backbones to relieve the switch congestion. In this case, the RCG POTS interface – 40 can be used solely as a lifeline connection, or when the 802.11 wireless LAN in which the RCG resides is congested. With this implementation, voice calls will be carried over the wireless interface as opposed to the POTS interface on which they would normally be carried. Also, the RCG modem - 41 connected to the POTS interface – 40 would

not need to maintain a constant always on connection, it could be a dialed on demand connection and thereby relieving the local class 5 office switch.

Bouncing or Hopping traffic:

[0081] Refer to figure 9. In areas where the density of RCGs is less than the arbitrary number of 32 RCGs within its transmission radius for multilink connections, the RCGs can bounce, or relay, the data traffic between each other to reach other RCGs located farther than the RCGs wireless transmission area. Since the POTS data rate of 56Kbps is miniscule as compared to the 11/54 Mbps 802.11b/g transfer rate, the overhead associated with bouncing data traffic is not a concern. This will allow for the fastest possible data transfer across POTS lines for every RCG user.

[0082] RCGs can also communicate directly with an 802.11a Neighborhood Access Point - 93 device (such as that used in WI-FI networks) that is connected to a data network connection such as a DS3 or OC3 service via a Metro Access Point. Where an RCG has access to a broadband device such as an 802.11a type device, either directly or via a bounced signal across several RCGs, a direct high speed data connection can be used to transfer large files. When an RCG is communicating directly with an 802.11a device, data transfer speeds up to the full 11/54 Mbps can be achieved to speed video transfers for such applications as video on demand.

RCG Router Tables:

[0083] The RCG continually monitors its 802.11b/g wireless interface for transmissions from other RCGs or 802.11a Access Points as well as other 802.11b/g wireless devices within the residence. The RCG provides a means to identify and store the addresses of other RCGs and Neighborhood Access Points within its local area. The RCG creates routing tables, much like an IP router device, so that it knows which RCGs are local to itself, and how many hops will be needed to reach the maximum of 32 POTS lines or a Neighborhood Access Point. The RCG can then determine how to route its data transfers to gain the most efficient routing and network efficiencies.

[0084] The router tables in each RCG are continually updated by monitoring the 802.11 frames received from other RCGs or Network Access Points. The number of entries in the router tables can be limited to an arbitrary number of other RCGs but contains room for at least 32

entries. For each RCG listed in the router tables, certain information is kept such as signal strength, if the other device is another RCG or some other 802.11 compatible device, if the other RCG is reachable directly, or if it is accessed via another RCGs by hopping. In the case there RCG hopping is used, the address of each RCG that is hopped is stored in the routing table. When an RCG is first powered up, it monitors its 802.11 interface for a period of time to identify other RCGs or Neighborhood Access Point devices, or other Wi-Fi devices within its transmission area. When first deployed, the RCG will not have any entries in its routing tables. This condition can persist for some time, until another RCG or other 802.11 device is deployed within the former RCGs transmission range. In this state, the RCG can be used for wireless networking within the residence, connecting the local computer to the RCGs always on Internet connection as well as to other wireless devices within the home. When an 802.11 frame is received by an RCG that contains the address of an RCG whose address does not exist in the receiving RCG's router table (as is the case when a new RCG is first installed), the receiving RCG polls the transmitting RCG for its router table information. The received router information is reviewed for other RCG's or other 802.11 compatible devices that are not listed in its own router tables. A decision is made to use some, all, or none of the new route information in its router tables. There is no real limitation on the number of RCGs that can be "hopped" in order to build a complete router table or to reach many more RCGs, Neighborhood Access Points or Metro Access Points.

Equipment and Services Priorities:

[0085] Just as voice has priority over data transfers within a single RCG, local RCG traffic (both voice and data) has priority over remote transfer requests by other RCGs. If an RCG is currently using all of its available POTS bandwidth for its own purposes, it will deny any remote RCG's request for access to its POTS line. Only when there is available POTS bandwidth available will an RCG allow other RCGs access to its POTS line. It will, however, bounce the traffic to other RCGs within its transmission area if requested by the originating RCG. In this case, available wireless bandwidth of any RCG (that amount of the 11/54 Mbps not currently used for its own traffic) can be used to relay traffic between other RCGs that do have available POTS bandwidth. In the same fashion, excess wireless bandwidth of any RCG can be by other RCGs to access high speed backbones via Neighborhood or Metro Access Points.

[0086] RCGs can also access other 802.11 wireless networks (Wi-Fi for example) that happen to be deployed within their service areas. In this case, access agreements between the RCG service provider and the wireless network operator must be put into place. By remaining compatible with the 802.11 standards, the RCGs capabilities will benefit by the growing WI-FI network infrastructure, an the WI-FI network will benefit by the RCG network.

Voice Over POTS and Data Over Wireless:

[0087] Normally, all communications from an RCG will be transmitted over the local POTS line, or a multilink bundle made up of many remote RCG POTS lines. By default, all voice calls will be carried over the "local" POTS line – 40 to alleviate security/privacy concerns and only "data" will be transmitted via multilink bundles. Generally, since voice "data" is compressed locally by the RCG, the required bandwidth for a voice call is so small that it does not make for a good choice to route it over a multilink connection. However, it is possible to route voice calls entirely over another RCGs POTS line. It is also possible for an RCG to transmit voice data over POTS lines connected to other RCGs via multilink bundles or directly over its 802.11 wireless interface, and since the 802.11 specification does support the Wired Equivalent Privacy (WEP) specification, a secured connection can be made in this fashion.

Remote Upgradability

[0088] The RCG is designed with the capability for unassisted remote upgradability. The remote upgradability feature is not the same as the storing of operational parameters during the configuration process. This feature allows the system operator to add new features as they are developed, or to correct bugs remotely and without the users knowledge or involvement. This feature can also be used to install a complete new operating system (OS).

[0089] Key to the successful implementation of the Remote upgradability feature is that a completely fail-safe routine be implemented that can recover from any failure during the upgrade process and to re-initiate and monitor the upgrade process until it has successfully completed. This process is carried out by loading a boot loader program that is stored in non-volatile memory. The boot loader program provides the minimum capabilities to start up the RCG and connect into the system operator's diagnostics network to download fixes or a new OS. The boot loader program is technically part of the OS, but it cannot be over written by any upgrade

process. It is responsible for detecting failure of the RCG and initiating a connection to the diagnostics network. Once connected with the diagnostics network, the boot loader will provide diagnostics support to help correct the RCG's operation and reconfigure the OS.

CLAIMS

What is claimed is:

1. A communications device that is capable of providing broadband communications services, comprising:

a wireless interface for connecting wirelessly to at least two remote communications devices; and

a processor that requests the at least two remote communications devices to assist in transferring data and aggregates the data from the at least two remote communications devices.

2. The communications device of claim 1, wherein the processor allocates the at least two remote communications device into a multilink group transfer.

3. The communications device of claim 1, wherein the processor creates and updates routing tables with the at least two remote communications devices.

4. The communications device of claim 3, wherein the processor uses the routing tables to determine routing efficiencies based on at least one of quality of service, total connection speed of the at least two remote communication devices, a number of data hops, available unused bandwidth of the at least two remote communication devices, wireless transfer speeds and signal strength between the at least two remote communication devices.

5. The communications device of claim 1, wherein the processor determines how many remote communications devices are located in a transmission range.

6. The communications device of claim 1, wherein the processor sends a request to the at least two remote communications devices for requesting use of a portion of unused bandwidth of the at least two remote communications devices, receives a response from the at least two remote communications devices, the response containing information about the portion of unused bandwidth, sends control information to the at least two remote communications device for participating in a multilink connection, and receiving packets from the at least two remote communications devices.

7. A communications device that is capable of providing broadband communications services, comprising:

at least one connection for connecting to a network;

at least one wireless interface for connecting wirelessly to remote communications devices; and

a processor that requests the remote communications devices to transfer portions of data, and combines and sequences the portions of the data from the remote communications devices with second data transferred from the at least one connection to increase data bandwidth for the communications device.

8. A communications device that is capable of providing broadband communications services, comprising:

a wireless interface for connecting wirelessly to at least two wireless devices; and

a processor that requests the at least two wireless devices to assist in transferring data and aggregates the data from the at least two wireless devices.

ABSTRACT

The Residential Communications Gateway (RCG) is a broadband communications device that combines all voice, data and video communications to and from a typical residence or small business for transmission over a single, or a plurality of Plain Old Telephone Service (POTS) lines separately or in conjunction with, a wireless broadband backbone. The RCG does this by employing packetized data with Voice over Internet Protocol (VoIP) technologies combined with RF communications technologies. A key consideration to the design of the RCG is that no additional or special transmission equipment must be installed at the Central Office or anywhere else in the network to enable new calling features provided by the RCG as is the case with DSL and Cable systems. By eliminating the requirement for costly infrastructure enhancements, ubiquitous high speed communications and services can be deployed to every POTS subscriber.

RED HAT 1012
RED HAT V. CAS

16 – CPU USB INTERFACE
12 – USB PORT
11 – ETHERNET PORT
13 – USB DRIVER
21 – DISPLAY AND KEY DRIVERS
10 – DISPLAY AND KEYPAD
15 – USB VIDEO INPUT
17 –ETHERNET DRIVER
ETHERNET DRIVER
USB DRIVER
VIDEO CAMERA
14 – VIDEO CAMERA
DISPLAY AND KEY PAD
22 – SPEAKERPHONE DRIVERS
22 – DISPLAY AND KEYPAD CABLES
18 – CPU E-NET I/F
42 – CPU TO MODEM/DAA I/F
41 – MODEM/DAA
23 – MICROPHONE CABLE
19 – MAIN CPU
V.90/V.92 MODEM/DAA
40 – INCOMING POTS PORT
20 – CPU DISPLAY & KEYPAD I/F
DISPLAY & KEY PAD I/F
MAIN SYSTEM PROCESSOR
37 – LIFELINE BYPASS ROUTING
24 – MICROPHONE
32 – DSP SPKR PHONE I/F
43 – CPU LIFELINE INT
SPEAKERPHONE I/F
LIFELINE FAIL-SAFE LOGIC
25 – SPEAKER
33 – DSP ENGINE
39 – LIFELINE FAILSAFE LOGIC
28 – POTS 3 PORT
CODECS AND SLICS
26 – SPKR CABLE
VOICE PROC-DSP AND IP PACKETIZATION
POWER CIRCUITRY
38 – POWER CONDITIONING AND MONITORING
29 – POTS 2 PORT
3 TELEPHONE INTERFACES
35 – 802.11B I/F
36 – 802.11B/G MODULE
30 – POTS 1 PORT
IRDA I/F
WIFI MODULE
37 – LIFELINE BYPASS ROUTING
27 – SLIC/CODEC INTERFACE
31 – DSP AUDIO INTERFACE
34 – CPU/DSP COMM PATH
8 – IRDA PORT
9 – 802.11B INTERFACE
10 – IRDA Interface

FIG. 1

58 – PSTN NETWORK
59 – IXC NETWORK
SOFTSWITCH, SIP PROXY SERVER AND REGISTRAR
51 – SOFTSWITCH, SIP PROXY SERVER AND REGISTRAR
52 – IP BASED NETWORK ROUTE
53 – IP BASED NETWORK ROUTE
49 – IXC B MODEM/ROUTER
46 – IXC A INTEROFFICE TRUNK
IXC A MODEM BANK/ROUTE
IXC B MODEM BANK/ROUT
54 – IXC B INTEROFFICE TRUNK
45 – USER A POTS LINE
57 – IP BASED RTP STREAM
55 – USER B POTS LINE
RCG
CLASS 5 OFFICE A
48 – IXC A MODEM/ROUTER
CLASS 5 OFFICE B
RCG
50 – CLASS 5 OFFICE B
44 – USER A RCG
47 – CLASS 5 OFFICE A
56 – USER B RCG

FIG. 2

59 – IXC NETWORK
58 – PSTN NETWORK
51 – SOFTSWITCH, SIP PROXY SERVER AND REGISTRAR
SOFTSWITCH, SIP PROXY SERVER AND REGISTRAR
52 - IP BASED NETWORK ROUTE
53 - IP BASED MEDIA GATEWAY CONTROL
46 – IXC A INTEROFFICE TRUNK
60 – IXC MEDIA GATEWAY
54 – IXC B INTEROFFICE TRUNK
45 – USER A POTS LINE
IXC A MODEM BANK/ROUTER
57 – IP BASED RTP STREAM
MEDIA GATEWAY - SIP BASED
55 – USER B POTS LINE
RCG
CLASS 5 OFFICE A
48 – IXC A MODEM/ROUTER
CLASS 5 OFFICE B
50 – CLASS 5 OFFICE B
44 – USER A RCG
47 – CLASS 5 OFFICE A
61 – USER B TELEPHONE

FIG. 3

72 - DISPLAY
Display (optionally a touch screen)
71-75 – PROGRAMMABLE FUNCTION KEYS
77-79 – LINE SELECT BUTTONS
76 – NUMBER KEYPAD
70 – RCG BASE
25 –SPEAKER
1
2
3
4
5
6
7
8
9
*
0
#
46 – WEIRLESS KEYBOARD PORT - IRDA
24 –MICROPHONE
80-82 – LINE ACTIVE LED'S

FIG. 4

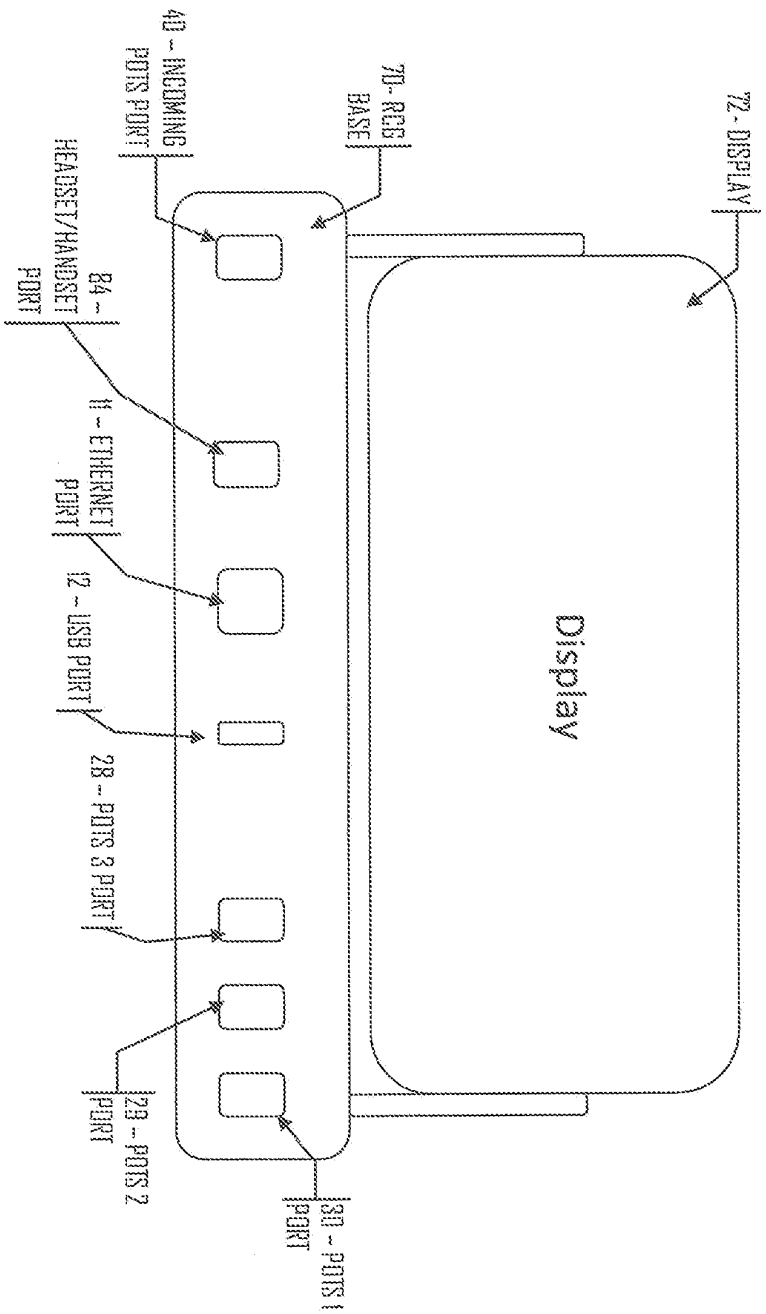


FIG. 5

RED HAT 1012
RED HAT V. CAS

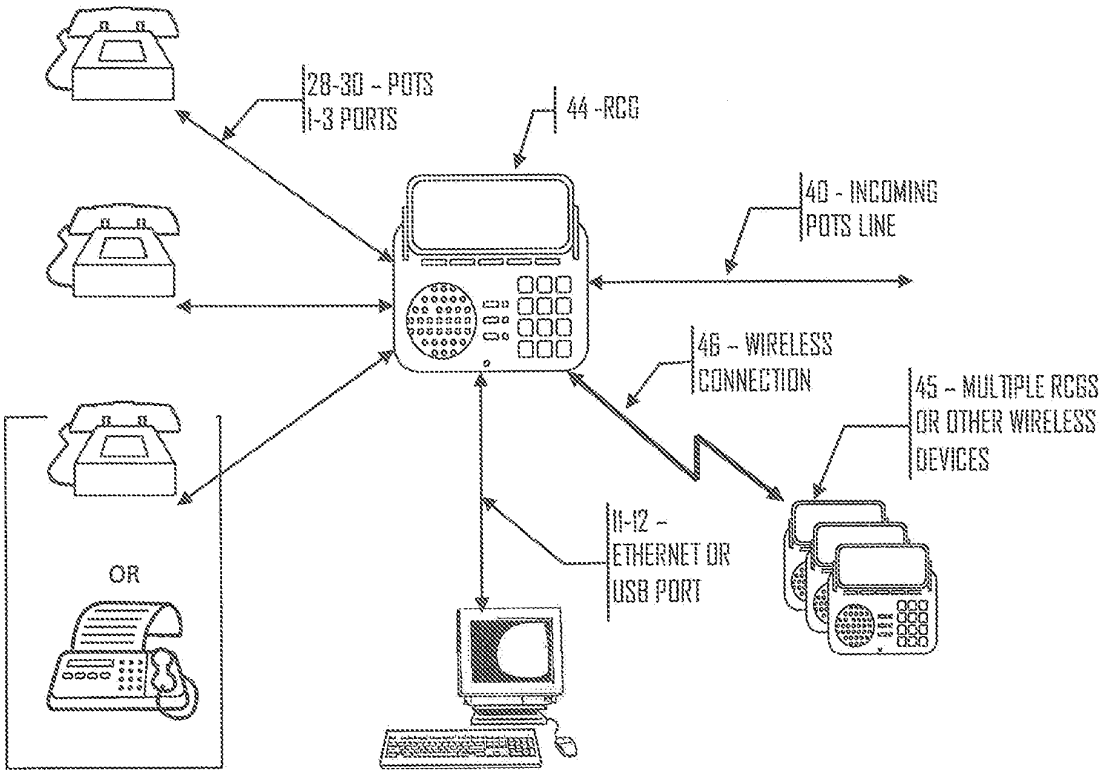


FIG. 6

RED HAT 1012
RED HAT V. CAS

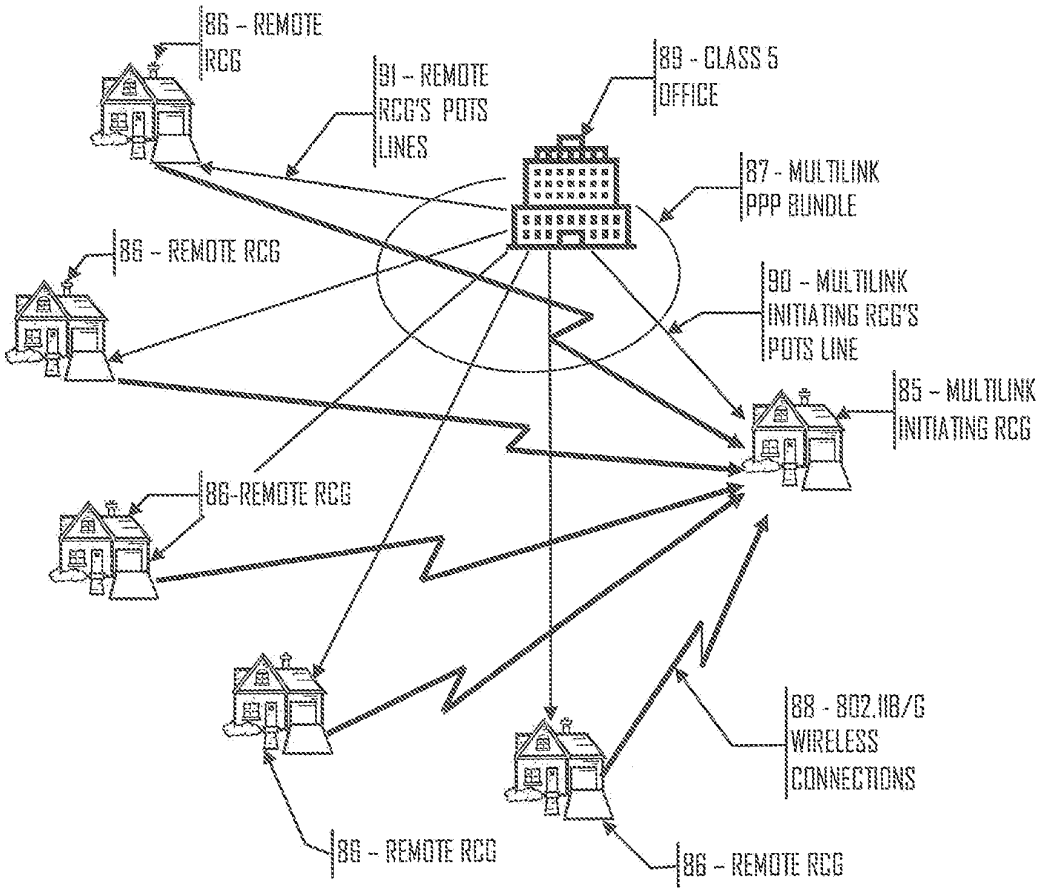


FIG. 7

RED HAT 1012
RED HAT V. CAS

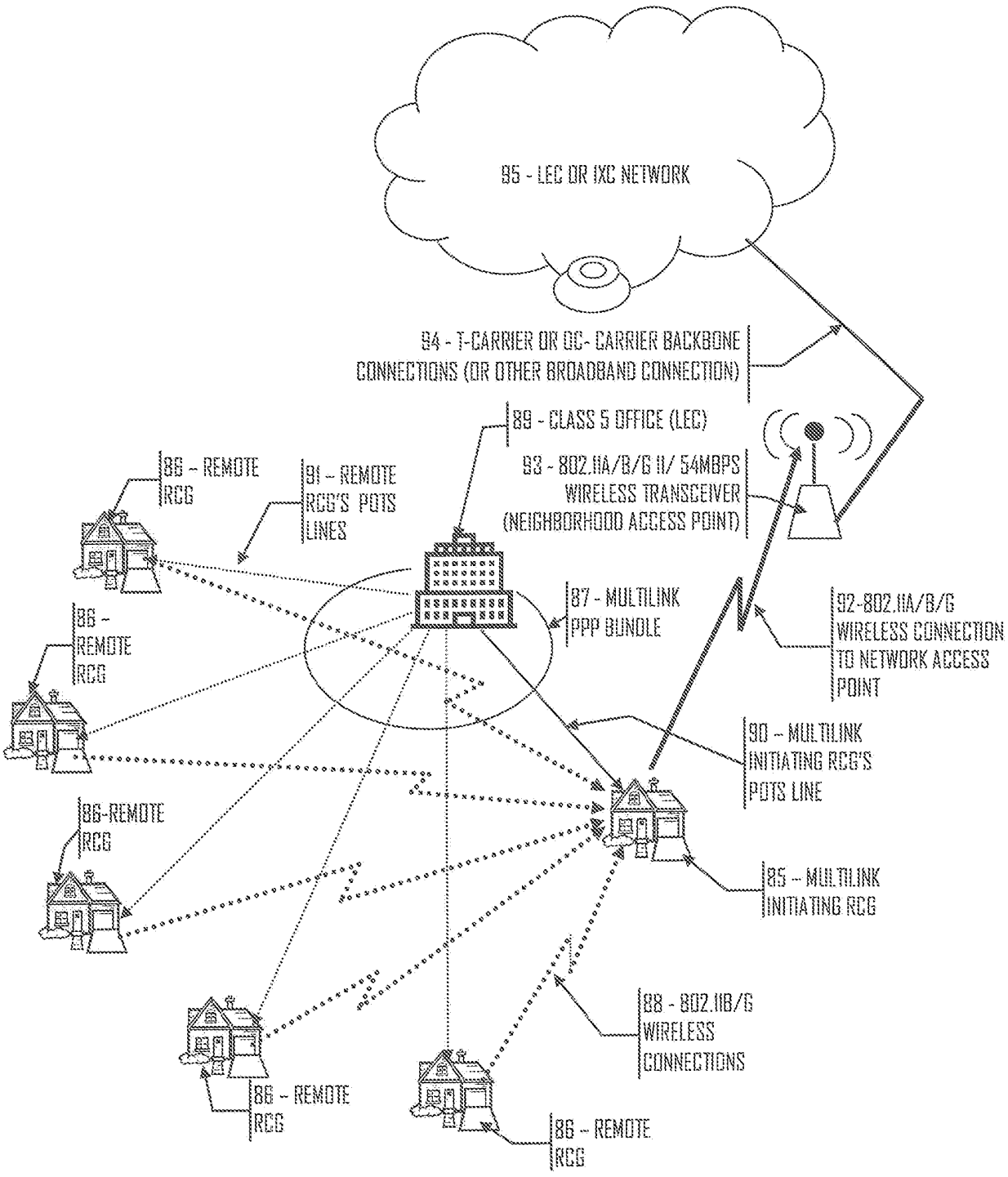


FIG. 8

RED HAT 1012
RED HAT V. CAS

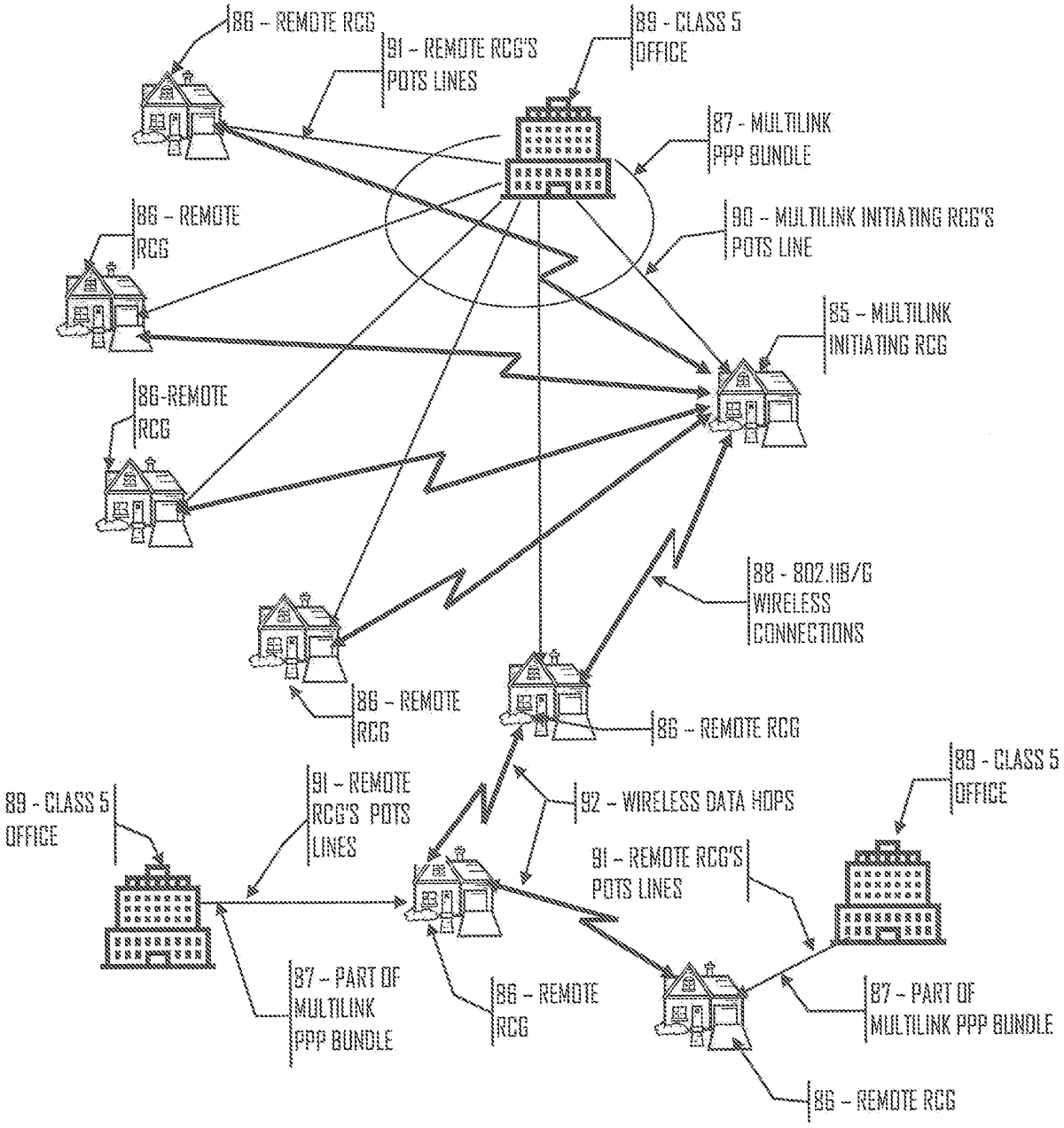


FIG. 9

RED HAT 1012
RED HAT V. CAS

PTO/SB/01A (01-09)
Approved for use through 06/30/2010. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	A BROADBAND COMMUNICATIONS DEVICE

As the below named inventor(s), I/we declare that:

This declaration is directed to:

☒ The attached application, or

☐ Application No. _____ filed on _____

☐ As amended on _____ (if applicable);

I/we believe that I/we am/are the original and first inventor(s) of the subject matter which is claimed and for which a patent is sought;

I/we have reviewed and understand the contents of the above-identified application, including the claims, as amended by any amendment specifically referred to above;

I/we acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me/us to be material to patentability as defined in 37 CFR 1.56, including for continuation-in-part applications, material information which became available between the filing date of the prior application and the national or PCT international filing date of the continuation-in-part application.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

All statements made herein of my/our own knowledge are true, all statements made herein on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001, and may jeopardize the validity of the application or any patent issuing thereon.

FULL NAME OF INVENTOR(S)

Inventor one: ERIC M. DELANGIS Date: 10/19/2009

Signature: _____ Citizen of: USA

Inventor two: _____ Date: _____

Signature: _____ Citizen of: _____

☐ Additional inventors or a legal representative are being named on _____ additional form(s) attached hereto.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 1 minute to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RED HAT 1012
RED HAT V. CAS

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 13/531,294

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	95		N/A	
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	310		N/A	
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	125		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	8 minus 20 =	*	x 30 =	0.00	OR		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3 minus 3 =	*	x 125 =	0.00			
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			0.00			
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				0.00			
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	530		TOTAL	

APPLICATION AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
Application Size Fee (37 CFR 1.16(s))									
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
Application Size Fee (37 CFR 1.16(s))									
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/531,294	06/22/2012	Eric M. Delangis	ED 1040

CONFIRMATION NO. 9530

64828
Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

POA ACCEPTANCE LETTER

OC000000055180196

Date Mailed: 07/05/2012

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 06/22/2012.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/tnguyen/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
13/531,294	06/22/2012	2614	530	ED 1040	8	3

CONFIRMATION NO. 9530

64828
Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

FILING RECEIPT

OC000000055180073

Date Mailed: 07/05/2012

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Eric M. Delangis, Huntsville, AL;

Power of Attorney: The patent practitioners associated with Customer Number 64828

Domestic Priority data as claimed by applicant

This application is a CON of 12/581,852 10/19/2009
which is a CON of 10/686,375 10/14/2003 PAT 7606156 *
which claims benefit of 60/418,521 10/15/2002
(*)Data provided by applicant is not consistent with PTO records.

Foreign Applications (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.)

If Required, Foreign Filing License Granted: 07/03/2012

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 13/531,294**

Projected Publication Date: 10/11/2012

Non-Publication Request: No

Early Publication Request: No
**** SMALL ENTITY ****

RED HAT 1012
RED HAT V. CAS

Title

BROADBAND COMMUNICATIONS DEVICE

Preliminary Class

379

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as

RED HAT 1012
RED HAT V. CAS

set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage, facilitate, and accelerate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/531,294	06/22/2012	Eric M. Delangis	ED 1040

CONFIRMATION NO. 9530

64828
Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

PUBLICATION NOTICE



Title:BROADBAND COMMUNICATIONS DEVICE

Publication No.US-2012-0258673-A1
Publication Date:10/11/2012

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently http://pair.uspto.gov/. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

To: CLAYMCGURK@COX.NET,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

Oct 12, 2012 05:19:49 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	NTC.PUB	10/11/2012	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

64828 7590 11/29/2012

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

EXAMINER	
CHO, HONG SOL	

ART UNIT	PAPER NUMBER
2467	

NOTIFICATION DATE	DELIVERY MODE
11/29/2012	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

CLAYMCGURK@COX.NET

Office Action Summary	Application No.	Applicant(s)	
	13/531,294	DELANGIS, ERIC M.	
	Examiner	Art Unit	
	HONG CHO	2467	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *1-8* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1-8* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

10)☒ The specification is objected to by the Examiner.

11)☒ The drawing(s) filed on *22 June 2012* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

12)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Specification

1. The disclosure is objected to because of the following informality:

The Applicant is required to provide the status (if patented or abandoned) of the applications, 10/686375 and 12/581852.

Claim Rejections - 35 USC § 112, First paragraph

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claim 2 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter, which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Re claim 2, the specification fails to explain how to allocate devices into a multilink group transfer.

Claim Rejections - 35 USC § 112, Second paragraph

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

RED HAT 1012
RED HAT V. CAS

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claim 2 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Re claim 2, it is not clear what is meant by allocating the device into a multilink group transfer. The specification fails to describe what the multilink group is.

Double Patenting

6. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

RED HAT 1012
RED HAT V. CAS

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

7. Claims 1, 2 and 5-7 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1, 4 and 7 of U.S. Patent No. 8228801. Although the conflicting claims are not identical, they are not patentably distinct from each other because:

Claim 6 of the instant application merely broadens the scope of the claim 1 of the Patent by eliminating the element and its function of the claim (selects the at least one remote communications device with unused bandwidth). It has been held that the omission of an element and its function is an obvious expedient if the remaining elements perform the same function as before. *In re Karlson*, 136 USPQ 184 (CCPA).

Claims 2 and 5 correspond substantially to patent claims 4 and 7.

Claim 7 of the instant application merely broadens the scope of the claim 10 of the Patent by eliminating the element and its function of the claim (sends a request to the remote communications devices for requesting use of a portion of unused bandwidth of each of the remote communications devices, receives a response from each of the remote communications devices, the response containing information about the portion of unused bandwidth for each of the remote communications devices, selects a group of the

RED HAT 1012
RED HAT V. CAS

remote communications devices with a larger, combined, unused bandwidth, sends control information to the selected remote communications devices for participating in a multilink transfer, and receiving packets from the selected remote communications devices). It has been held that the omission of an element and its function is an obvious expedient if the remaining elements perform the same function as before. *In re Karlson*, 136 USPQ 184 (CCPA).

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102(e) that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9. Claims 1, 2 and 6-8 are rejected under 35 U.S.C. 102(e) as being anticipated by Barany et al (US 7068623), hereinafter referred to as Barany.

Re claims 1, 7 and 8, Barany discloses a communication device with a wireless interface for connecting to wireless devices and requests the remote communications devices to transfer portions of data, and combines and sequences the portions of the data

RED HAT 1012
RED HAT V. CAS

from the remote communications devices with second data transferred from the at least one connection to increase data bandwidth for the communications device (figure 2).

Re claim 2, as best understood, Barany discloses multiplexing of traffic transmitted by first and second mobile stations.

Re claim 6, Barany discloses requesting use of a portion of unused bandwidth of the at least two remote communications devices, receiving a response containing information about the portion of unused bandwidth (column 5, lines 1-12; lines 19-26), sending control information to the at least two remote communications device for participating in a multilink connection, and receiving packets from the at least two remote communications devices (figure 3).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 3 and 4 are rejected under 35 U.S.C. 103(a) as being unpatentable over Barany in view of Cain et al (US 20030202468), hereinafter referred to as Cain.

Re claims 3 and 4, Barany discloses all of the limitations of the base claim, but fails to disclose using routing tables to determine routing efficiencies based on one of

RED HAT 1012
RED HAT V. CAS

QoS, connection speed, unused bandwidth, hops, wireless transfer speed and signal strength. However, Cain discloses generating QoS metrics for each potential route selection. The QoS metrics may include at least one of available bandwidth, hop count (paragraph [0020]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Barany with the teaching of Cain for the benefit of providing efficient routing of data packets based on QoS requirement.

Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Barany in view of Leung et al (US 20030063591), hereinafter referred to as Leung.

Re claim 5, Barany discloses all of the limitations of the base claim, but fails to disclose determining how many remote communications devices are located in a transmission range. However, Leung discloses building a multicast tree based on the transmission range information (paragraph [0083]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Barany with the teaching of Leung for the benefit of efficient utilization of network resources by multicasting data packets within a transmission range.

Conclusion

RED HAT 1012
RED HAT V. CAS

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hong Cho whose telephone number is 571-272-3087. The examiner can normally be reached on Mon-Fri during 7 am to 4 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hassan Phillips can be reached on 571-272-3940. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Hong Cho/

Primary Examiner, Art Unit 2467

RED HAT 1012
RED HAT V. CAS

Notice of References Cited	Application/Control No. 13/531,294	Applicant(s)/Patent Under Reexamination DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,307,839 B1	10-2001	Gerszberg et al.	370/235
*	B	US-2003/0063591 A1	04-2003	Leung et al.	370/338
*	C	US-2003/0202468 A1	10-2003	Cain et al.	370/229
*	D	US-2006/0079274 A1	04-2006	Gallagher et al.	455/552.1
*	E	US-7,068,623 B1	06-2006	Barany et al.	370/329
*	F	US-7,257,106 B2	08-2007	Chen et al.	370/338
*	G	US-7,308,279 B1	12-2007	Zweig, Jonathan M.	455/522
*	H	US-7,385,924 B1	06-2008	Riddle, Guy	370/235
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 9530

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
13/531,294	06/22/2012 **RULE**	370	2467	ED 1040

APPLICANTS
Eric M. Delangis, Huntsville, AL;

**** CONTINUING DATA ************************
This application is a CON of 12/581,852 10/19/2009 PAT 8,228,801
which is a CON of 10/686,375 10/14/2003 PAT 7,606,156 *
which claims benefit of 60/418,521 10/15/2002
(*)Data provided by applicant is not consistent with PTO records.

**** FOREIGN APPLICATIONS ************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED **** ** SMALL ENTITY **
07/03/2012

Foreign Priority claimed ☐ Yes ☑ No 35 USC 119(a-d) conditions met ☐ Yes ☑ No Verified and Acknowledged /HONG SOL CHO/ Examiner's Signature	☐ Met after Allowance Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		AL	9	8	3

ADDRESS
Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

TITLE
BROADBAND COMMUNICATIONS DEVICE

FILING FEE RECEIVED		
530	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

BIB (Rev. 05/07).

RED HAT 1012
RED HAT V. CAS

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	Art Unit
	HONG CHO	2467

SEARCHED			
Class	Subclass	Date	Examiner
370	230-235,278,282,328,338,468,477	11/26/2012	hc

SEARCH NOTES		
Search Notes	Date	Examiner
EAST text search	11/26/2012	hc

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	11/26/2012								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S29	3	"12581852"	US-PGPUB; USPAT	OR	OFF	2012/11/26 09:38
S30	5	10/686375	US-PGPUB; USPAT	OR	OFF	2012/11/26 09:39
S31	9	((ERIC) near2 (DELANGIS)).INV.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/11/26 09:41
S32	37466	370/230-235,278,282,328,338,468,477.ccls.	US-PGPUB; USPAT	OR	OFF	2012/11/26 10:25
S37	1	"13531294"	US-PGPUB; USPAT	OR	OFF	2012/11/26 10:30
S48	809	(aggregat$3 or combin$3 or add$3) near (data or message or inforamtion) with (wireless or mobile) adj (devices or station$1)	US-PGPUB; USPAT	OR	OFF	2012/11/26 10:50
S50	544	(aggregat$3 or combin$3) near (data or message or inforamtion) with (wireless or mobile) adj (devices or station$1)	US-PGPUB; USPAT	OR	OFF	2012/11/26 10:51
S51	67	S50 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2012/11/26 10:52
S53	195	S32 and (multiplex$3 or aggregat$3 or combin$3) near (traffic or data or message$1 or inforamtion or packet$1 or frame$1) with (wireless or mobile) adj (devices or station$1)	US-PGPUB; USPAT	OR	OFF	2012/11/26 11:13
S54	26	S53 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2012/11/26 11:14
L5	25	(select$3 or determin$3 or choos$3) with rout$3 near path with qos same hop	US-PGPUB; USPAT	OR	OFF	2012/11/26 13:20
L11	41	(bs or ap)with (select$3 or determin$3 or choos$3) with rout$3 with hop	US-PGPUB; USPAT	OR	OFF	2012/11/26 13:28
L15	30	(select$3 or determin$3 or choos$3) with rout$3 with qos same hop same bandwidth	US-PGPUB; USPAT	OR	OFF	2012/11/26 13:30
L18	37	multicast$3 with range near transmission	US-PGPUB; USPAT	OR	OFF	2012/11/26 13:48
L20	7	multicast$3 with range near transmission same tree	US-PGPUB;	OR	OFF	2012/11/26 13:58

			USPAT			

11/26/2012 2:52:31 PM
C:\Users\hcho\Documents\EAST\Workspaces\13531294.wsp

To: CLAYMCGURK@COX.NET,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

Nov 29, 2012 05:22:36 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	CTNF	11/29/2012	ED 1040
	892	11/29/2012	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

AMENDMENT

In response to the Office Action issued on November 29, 2012 ("Office Action"), please reconsider the above-identified application in light of the following amendments to the specification, claims and accompanying remarks. The shortened period for response was to expire on February 29, 2013 (which does not exist). A petition for extension of time is requested for two months, thus extending the period for response to April 29, 2013. This Amendment is being filed on April 29, 2013.

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION

I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: April 29, 2013

By: Clay McGurk
Clay McGurk

RED HAT 1012
RED HAT V. CAS

IN THE SPECIFICATION

Please replace paragraph 1 of the specification with the following:

[0001] This application is a continuation of U.S. Patent Application Serial No. 12/581,852 filed October 19, 2009, now U.S. Patent 8,228,801, which is a continuation of U.S. Patent Application Serial No. 10/686,375 filed October 14, ~~2005~~ 2003, now U.S. Patent 7,606,156, which claims priority to U.S. Provisional Application Serial No. 60/418,521, filed October 15, 2002, all which are incorporated by reference herein.

RED HAT 1012
RED HAT V. CAS

THE CLAIMS

This listing of claims will replace all prior versions and listings of the claims in the application:

1. (Currently Amended) A communications device that is capable of providing broadband communications services, comprising:

a wireless interface for connecting wirelessly to at least two remote communications devices; and

a processor that requests the at least two remote communications devices to assist in ~~transferring~~ downloading portions of data, receives the portions of the data from the at least two remote communication devices and aggregates the portions of the data downloaded from the at least two remote communications devices to create a data stream for the communication device.

2. (Currently Amended) The communications device of claim 1, wherein the processor allocates the at least two remote communications device into a multilink ~~group transfer~~ communication.

3. (Original) The communications device of claim 1, wherein the processor creates and updates routing tables with the at least two remote communications devices.

4. (Original) The communications device of claim 3, wherein the processor uses the routing tables to determine routing efficiencies based on at least one of quality of service, total connection speed of the at least two remote communication devices, a number of data hops, available unused bandwidth of the at least two remote communication devices, wireless transfer speeds and signal strength between the at least two remote communication devices.

5. (Currently Amended) The communications device of claim 1, wherein the processor determines ~~how many~~ a number of remote communications devices that are located in a transmission range with the communications device.

RED HAT 1012
RED HAT V. CAS

6. (Currently Amended) The communications device of claim 1, wherein the processor sends a request to each of the at least two remote communications devices for requesting use of a portion of unused bandwidth of the at least two remote communications devices, receives a response from each of the at least two remote communications devices, the response containing information about availability of the portion of unused bandwidth, sends control information to the at least two remote communications device so that the at least two remote communication devices receive a set of packets ~~participating in a multilink connection~~, and receiving the set of packets from the at least two remote communications devices.

7. (Currently Amended) A communications device that is capable of providing broadband communications services, comprising:

at least one connection for connecting to a network;

a at ~~least one~~ wireless interface for connecting wirelessly to one or more remote communications devices; and

a processor that requests one or more of the remote communications devices to download ~~transfer~~ portions of data, receives the portions of data downloaded by the remote communication devices and combines and sequences the portions of the data from the remote communications devices with second data transferred from the at least one connection to increase data transfer speed ~~bandwidth~~ for the communications device.

8. (Currently Amended) A communications device that is capable of providing broadband communications services, comprising:

a wireless interface for connecting wirelessly to ~~at least two wireless devices~~ a first wireless device and a network; and

a processor that requests the ~~at least two wireless devices~~ first wireless device to download a first portion of ~~assist in transferring~~ data, receives a second portion of data from the network, receives the first portion of data from the first wireless device and aggregates the first portion of data and the second portion of data to create a data stream ~~from the at least two wireless devices~~ that is downloaded to the communication device.

RED HAT 1012
RED HAT V. CAS

9. (New) The communication device of claim 7, where the network is a telephone network, the Internet, or a telecommunication system.

10. (New) The communication device of claim 8, where the network is a telephone network, the Internet, or a telecommunication system.

11. (New) The communication device of claim 8, wherein the wireless interface connects to a second wireless device, and

wherein the processor requests the second wireless device to download a third portion of data and aggregates the first portion of data, the second portion of data and the third portion of data to create the data stream that is downloaded to the communication device.

12. (New) The communication device of claim 8, wherein the wireless interface connects to a third wireless device, and

wherein the processor requests the third wireless device to download a fourth portion of data and aggregates the first portion of data, the second portion of data, the third portion of data and the fourth portion of data to create the data stream that is downloaded to the communication device.

13. (New) The communication device of claim 8, wherein the communication device is a cell phone or a mobile computer.

14. (New) The communication device of claim 8, wherein the first wireless device is a cell phone or a mobile computer.

15. (New) The communication device of claim 8, wherein the first wireless device is a wireless router.

16. (New) The communication device of claim 8, wherein the first wireless device connects to a wireless router.

RED HAT 1012
RED HAT V. CAS

REMARKS

Status of the Claims

Claims 1-8 were pending in this application before this Amendment. Claims 1-2 and 5-8 have been amended herein. Claims 9-16 are new claims and added herein. No new matter has been added. Thus, claims 1-16 are pending in this application pursuant to this Amendment.

Objection to the Specification

On page 2, paragraph 1 of the Office Action, the specification was objected to for not listing the current status of the prior pending applications. An amendment to the specification is requested herein. It is respectfully requested that the objection to the Specification be reconsidered and withdrawn.

Rejection of claims based on 35 U.S.C. §112

On pages 2-3 of the Office Action, claim 2 was rejected under 35 U.S.C. §112, first and second paragraphs. Although the specification describes multilink group transfer using different words, appropriate corrections to the claim has been made in this Amendment. It is respectfully requested that the rejections of claim 2 based on 35 U.S.C. §112 (first and second paragraphs) be reconsidered and withdrawn.

Double Patenting

On pages 3-5 of the Office Action, claims 1, 2 and 5-7 are rejected on the grounds of nonstatutory obviousness-type double patenting. Once the pending claims of the present application are indicated as being allowable over any cited prior art, a terminal disclaimer will most likely be filed to overcome any nonstatutory obviousness-type double patenting rejection that may exist at that time. It is respectfully requested that these rejections be postponed until the claims are indicated as being allowable except for any nonstatutory obviousness-type double patenting rejection or rejections.

35 U.S.C. §102 Rejections

On pages 5-6 of the Office Action, claims 1, 2 and 6-8 were rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent 7,068,623 to Barany *et al.* ("Barany"). "A claim is

RED HAT 1012
RED HAT V. CAS

anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987) (other citations omitted); *see also* M.P.E.P. §2131.

Barany does not describe or suggest a processor that requests at least two remote communication devices to assist in downloading portions of data, receives the portions of data and aggregates (claims 1 and 8), combines and sequences (claim 7) the portions of data as in claims 1, 7 and 8. In contrast to claims 1, 7 and 8, Barany describes a RAN (radio access network) 11 that multiplexes data from two mobile stations 100 and 102 on the same channel. (See Figs 1 and 2; 5:1-29; 6:8-15). Barany does not describe or suggest that the RAN 11 requests that the two mobile stations 100 and 102 download data for RAN 11. RAN 11 of Barany is the network responsible for providing voice and data channel communication to/from each of the mobile stations 100 and 102. In other words, Barany is trying to solve a different problem than what is being claimed in the present application.

For these reasons, Barany fails to describe claims 1, 7 and 8. Thus, claims 1, 7 and 8 are not anticipated by Barany and should be allowed over Barany. All the other claims depend on claims 1, 7 and 8 and should be similarly allowed due to their dependence on those allowable independent claims. Therefore, it is respectfully requested that claims 1-16 be allowed over Barany, and that the rejection under 35 U.S.C. §102 of claims 1, 2 and 6-8 as being anticipated by Barany be reconsidered and withdrawn.

35 U.S.C. §103 Rejections

On pages 6-7 of the Office Action, claims 3 and 4 were rejected under 35 U.S.C. §103(a) as being unpatentable over Barany in view of U.S. Publ. Appl. 2003-0202468 to Cain *et al.* ("Cain"). Claims 3 and 4 depend on claim 1, and claim 1 was shown to be allowable for the reasons discussed above. Thus, claims 3 and 4 are similarly allowable over Barany in view of Cain due to their dependence on an allowable independent claim. It is respectfully requested that the rejection of claims 3 and 4 under 35 U.S.C. §103 based on Barany in view of Cain be reconsidered and withdrawn.

On page 7 of the Office Action, claim 5 was rejected under 35 U.S.C. §103(a) as being unpatentable over Barany in view of U.S. Publ. Appl. 2003-0063591 to Leung *et al.* ("Leung").

RED HAT 1012
RED HAT V. CAS

Claim 5 depends on claim 1, and claim 1 was shown to be allowable for the reasons discussed above. Thus, claim 5 is similarly allowable over Barany in view of Leung due to its dependence on an allowable independent claim. It is respectfully requested that the rejection of claim 5 under 35 U.S.C. §103 based on Barany in view of Leung be reconsidered and withdrawn.

Conclusion

All rejections in the Office Action have been properly traversed. It is respectfully requested that all outstanding rejections be reconsidered and withdrawn. It is believed that a full and complete reply has been made in response to the Office Action, and, that as such, the present patent application is in a condition for allowance. The Examiner is always welcome to telephone the attorney of record at the telephone number provided below if such telephone discussion would help expedite the prosecution of the patent application. Prompt and favorable consideration of this Amendment is respectfully requested.

Best Regards,

Date: April 29, 2013

H C McGurk IV, Reg. No. 34,964
Attorney for Applicant

The Law Office of Clay McGurk
PO Box 1488
Orange, CA 92706
Phone: **949.395.3942**
Email: claymcgurk@cox.net

Electronic Patent Application Fee Transmittal

Application Number:	13531294
Filing Date:	22-Jun-2012
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Filer:	Harold C. McGurk
Attorney Docket Number:	ED 1040

Filed as Small Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 2 months with $0 paid	2252	1	300	300

RED HAT 1012
RED HAT V. CAS

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
		Total in USD ($)		**300**

RED HAT 1012
RED HAT V. CAS

Electronic Acknowledgement Receipt	
EFS ID:	15638361
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	29-APR-2013
Filing Date:	22-JUN-2012
Time Stamp:	15:41:06
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$300
RAM confirmation Number	2495
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)

RED HAT 1012
RED HAT V. CAS

1	Amendment/Req. Reconsideration-After Non-Final Reject	13531294_Amendment.pdf	104572 b0f6e07376ef1565bfc5dbc3cbf8acaa798f053b	no	8
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30201 5cbc806edebdb99e6c6e77559f3d9ea8b69b81a6	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	134773		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/531,294	Filing Date 06/22/2012	☐ To be Mailed

ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)		
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

		(Column 1)		(Column 2)	(Column 3)		
AMENDMENT	04/29/2013	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 16	Minus	** 20	= 0	X $40 =	0
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $210 =	0
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	0

		(Column 1)		(Column 2)	(Column 3)		
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/SHARAIN MORELAND/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

64828 7590 05/29/2013

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

EXAMINER	
CHO, HONG SOL	

ART UNIT	PAPER NUMBER
2467	

NOTIFICATION DATE	DELIVERY MODE
05/29/2013	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

CLAYMCGURK@COX.NET

Office Action Summary	Application No. 13/531,294	Applicant(s) DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 April 2013*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *1-16* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1-16* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some * c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Interim copies:

a)☐ All b)☐ Some c)☐ None of the: Interim copies of the priority documents have been received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. This office action is in response to the amendment filed on 04/29/13. Claims 1-16 are pending in the instant application.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 2 and 6-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Takahashi et al (US 6941151, "Takahashi") in view of Barany.

Re claims 1, 7 and 8, Takahashi discloses a communication device with a wireless interface for connecting to wireless devices (figure 1) and requesting the remote communications devices to download portions of data (column 9, lines 55-67), but fails to disclose combining and sequencing the portions of the data from the remote communications devices to create data stream for the communication device. However, Barany discloses multiplexing portions of the data from the remote communications

RED HAT 1012
RED HAT V. CAS

devices to create data stream for the communication device (figure 2). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Takahashi with the teaching of Barany for the benefit of providing efficient utilization of network resources by multiplexing data on a given channel.

Re claim 2, Takahashi discloses two remote communication devices in multilink communication with Internet or a telecommunication system (figure 1).

Re claim 6, Barany discloses requesting use of a portion of unused bandwidth of the at least two remote communications devices, receiving a response containing information about the portion of unused bandwidth (column 5, lines 1-12; lines 19-26), sending control information to the at least two remote communications device and receiving packets from the at least two remote communications devices (figure 3). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Takahashi with the teaching of Barany for the benefit of providing efficient utilization of network resources by using unused bandwidth for other communication session.

Re claims 9, 10, 13 and 14, Takahashi discloses a mobile station in communication over the Internet or a telecommunication system (figure 1).

Re claims 11 and 12, Takahashi discloses requesting the remote communications devices to download portions of data (column 9, lines 55-67), Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Takahashi so that a mobile station could request a particular

RED HAT 1012
RED HAT V. CAS

wireless device to download a certain part of data for the benefit of providing network service by considering a given network condition.

Re claims 15 and 16, Takahashi discloses all of the limitations of the base claim, but fails to disclose a first device being a router or connected to a wireless router. However, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Takahashi so that a first device is a router or connected to a wireless router for the benefit of supporting larger scale network environment.

Claims 3 and 4 are rejected under 35 U.S.C. 103(a) as being unpatentable over Takahashi in view of Barany and Cain et al (US 20030202468), hereinafter referred to as Cain.

Re claims 3 and 4, Takahashi discloses all of the limitations of the base claim, but fails to disclose using routing tables to determine routing efficiencies based on one of QoS, connection speed, unused bandwidth, hops, wireless transfer speed and signal strength. However, Cain discloses generating QoS metrics for each potential route selection. The QoS metrics may include at least one of available bandwidth, hop count (paragraph [0020]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Takahashi with the teaching of Cain for the benefit of providing efficient routing of data packets based on QoS requirement.

RED HAT 1012
RED HAT V. CAS

Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Takahashi in view of Barany and Leung et al (US 20030063591), hereinafter referred to as Leung.

Re claim 5, Takahashi discloses all of the limitations of the base claim, but fails to disclose determining how many remote communications devices are located in a transmission range. However, Leung discloses building a multicast tree based on the transmission range information (paragraph [0083]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Takahashi with the teaching of Leung for the benefit of efficient utilization of network resources by multicasting data packets within a transmission range.

Response to Arguments

4. Applicant's arguments with respect to claims have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

5. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the

RED HAT 1012
RED HAT V. CAS

shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hong Cho whose telephone number is 571-272-3087. The examiner can normally be reached on Mon-Fri during 7 am to 4 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hassan Phillips can be reached on 571-272-3940. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Hong Cho/

Primary Examiner, Art Unit 2467

RED HAT 1012
RED HAT V. CAS

RED HAT 1012
RED HAT V. CAS

Notice of References Cited	Application/Control No. 13/531,294	Applicant(s)/Patent Under Reexamination DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,941,151	09-2005	Takahashi et al.	455/502
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	Art Unit
	HONG CHO	2467

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
370	230-235,278,282,328,338,468,477	11/26/2012	hc

SEARCH NOTES		
Search Notes	Date	Examiner
EAST text search	11/26/2012	hc
updated search	5/21/2013	hc

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L6	1	"13531294"	US-PGPUB; USPAT	OR	OFF	2013/05/21 11:43
L12	1	(aggregat$3 or combin$3) with downloaded near(data or message or inforamtion) with (wireless or mobile) adj (devices or station$1)	US-PGPUB; USPAT	OR	OFF	2013/05/21 11:56
L13	194	request$3 with download$3 near(data or message or inforamtion or file) with (wireless or mobile) adj (device$1 or station$1)	US-PGPUB; USPAT	OR	OFF	2013/05/21 11:58
L14	19	l13 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/05/21 11:59
L15	4	"6941151"	US-PGPUB; USPAT	OR	OFF	2013/05/21 12:06
L18	80	(aggregat$3 or combin$3 or multiplex$3) near received near (packet or frame or cell or data or message or inforamtion) with sequenc$3	US-PGPUB; USPAT	OR	OFF	2013/05/21 12:40
L19	10	l18 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/05/21 12:41

5/ 21/ 2013 1:26:13 PM
C:\ Users\ hcho\ Documents\ EAST\ Workspaces\ 13531294.wsp

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	11/26/2012	05/21/2013							
	1	✓	✓							
	2	✓	✓							
	3	✓	✓							
	4	✓	✓							
	5	✓	✓							
	6	✓	✓							
	7	✓	✓							
	8	✓	✓							
	9		✓							
	10		✓							
	11		✓							
	12		✓							
	13		✓							
	14		✓							
	15		✓							
	16		✓							

To: CLAYMCGURK@COX.NET,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

May 29, 2013 05:22:10 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	CTFR	05/29/2013	ED 1040
	892	05/29/2013	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Mail Stop **AF**
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

AMENDMENT AFTER FINAL

In response to the final Office Action issued on May 29, 2012 ("Office Action"), please reconsider the above-identified application in light of the following amendments to the claims and accompanying remarks.

MAIL STOP AF

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION

I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: July 25, 2013

By: Clay McGurk
Clay McGurk

RED HAT 1012
RED HAT V. CAS

THE CLAIMS

This listing of claims will replace all prior versions and listings of the claims in the application:

1-16. (Canceled).

17. (New) A method for increasing bandwidth and data throughput comprising the steps of:

establishing a wireless connection between a first device and a second device, the first device being connected at a first location to a network, the second device being connected at a second location to the network, the second location being geographically separate from the first location, the first device and the second device being configured for accessing the network; and

routing data across the wireless connection thereby increasing bandwidth and data throughput at the first device than is achievable over the connection between the first device and the network.

18. (New) The method of claim 17, wherein the first location and the second location are physically isolated from each other.

19. (New) The method of claim 17, further comprising the step of aggregating the data received over the wireless connection and the data received over the connection at the first location.

20. (New) The method of claim 19, further comprising the step of creating a data stream from the data aggregated from the wireless connection and the connection at the first location.

21. (New) The method of claim 19, further comprising the step of creating a data file from the data aggregated from the wireless connection and the connection at the first location.

22. (New) The method of claim 17, further comprising the step of combining and sequencing the data received over the wireless connection and the data received over the connection at the first location.

23. (New) The method of claim 22, further comprising the step of creating a data stream from the data combined and sequenced from the wireless connection and the connection at the first location.

24. (New) The method of claim 22, further comprising the step of creating a data file from the data combined and sequenced from the wireless connection and the connection at the first location.

25. (New) The method of claim 17, further comprising the step of creating a first data stream from the data received over the wireless connection and a second data stream from the data received over the connection at the first location.

26. (New) The method of claim 17, further comprising the step of creating a first data file from the data received over the wireless connection and a second data file from the data received over the connection at the first location.

27. (New) The method of claim 17, wherein the wireless connection is established between a first wireless router at the first device and a second wireless router at the second device.

28. (New) The method of claim 17, wherein the network is one or a combination of a cable network, a telecommunication network, an optical network and a communications network.

29. (New) The method of claim 17, wherein the wireless connection is a cellular, satellite, Wi-Fi, WiMax or Bluetooth connection.

30. (New) A method for increasing bandwidth and data throughput comprising the steps of:

establishing a wireless connection between a first device and a second device, the first device being connected at a first location to a first network, the second device being connected at a second location to a second network, the second location being geographically separate from the first location, the first device being configured for accessing the first network, the second device being configured for accessing the second network; and

routing data across the wireless connection thereby increasing bandwidth and data throughput at the first device than is achievable over the connection between the first device and the first network.

31. (New) The method of claim 30, further comprising the step of aggregating the data received from the first network and the second network.

32. (New) The method of claim 31, further comprising the step of creating a data stream from the data aggregated from the first network and the second network.

33. (New) The method of claim 31, further comprising the step of creating a data file from the data aggregated from the first network and the second network.

34. (New) The method of claim 30, further comprising the step of creating a first data file from the data received from the first network and a second data file from the data received from the second network.

35. (New) A device comprising:

an interface for establishing a network connection at a first location to a network;

a wireless interface for establishing a wireless connection to a remote communications device, the remote communications device being connected at a second location to the network, the second location being geographically separate from the first location; and

a processor that requests the remote communications device to assist in routing data over the wireless connection, receives the data over the wireless connection thereby increasing communications bandwidth and data throughput than is achievable over the network connection.

36. (New) The device of claim 35, wherein the device is one of a modem, cellular phone, a router, a computer, hotspot device and a tablet.

37. (New) The device of claim 35, wherein the network connection is a wireless connection.

RED HAT 1012
RED HAT V. CAS

REMARKS

Status of the Claims

Claims 1-16 were pending in this application before this Amendment and are now canceled without prejudice or disclaimer. Claims 17-37 are new claims and are added in this Amendment. No new matter has been added. Thus, claims 17-37 are pending in this application pursuant to this Amendment.

35 U.S.C. §§ 102, 103 Rejections

On pages 2-4 of the Office Action, claims 1, 2 and 6-16 were rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent 6,941,151 to Takahashi *et al.* in view of U.S. Patent 7,068,623 to Barany *et al.* ("Barany"). On page 4 of the Office Action, claims 3 and 4 were rejected under 35 U.S.C. §103(a) as being unpatentable over Takahashi in view of Barany in view of U.S. Publ. Appl. 2003-0202468 to Cain *et al.* ("Cain"). On page 5 of the Office Action, claim 5 was rejected under 35 U.S.C. §103(a) as being unpatentable over Takahashi in view of Barany in view of U.S. Publ. Appl. 2003-0063591 to Leung *et al.* ("Leung"). These rejections are now moot because the rejections were based on claims that have been canceled without prejudice or disclaimer.

Conclusion

All rejections in the Office Action have been properly traversed. It is respectfully requested that all outstanding rejections be reconsidered and withdrawn. It is believed that a full and complete reply has been made in response to the Office Action, and, that as such, the present patent application is in a condition for allowance. The Examiner is always welcome to telephone the attorney of record at the telephone number provided below if such telephone discussion would help expedite the prosecution of the patent application.

Prompt and favorable consideration of this Amendment is respectfully requested.

Best Regards,

Date: July 25, 2013

H C McGurk IV, Reg. No. 34,964
Attorney for Applicant

The Law Office of Clay McGurk
PO Box 1488
Orange, CA 92706
Phone: **949.395.3942**
Email: claymcgurk@cox.net

RED HAT 1012
RED HAT V. CAS

Electronic Acknowledgement Receipt

EFS ID:	16419875
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	25-JUL-2013
Filing Date:	22-JUN-2012
Time Stamp:	17:20:52
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Response After Final Action	13531294_AAF_7-25-13.pdf	99446 3c95521b0ff09364ed004ae1337c7808d0c9001f	no	7
Warnings:					
Information:					

RED HAT 1012
RED HAT V. CAS

Total Files Size (in bytes):	99446

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/531,294	Filing Date 06/22/2012	☐ To be Mailed

ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT **07/25/2013**

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 21	Minus	** 20	= 1	X $40 =	40
Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $210 =	0
☐ Application Size Fee (37 CFR 1.16(s))						
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
					TOTAL ADD'L FEE	**40**

AMENDMENT

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))						
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
					TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/STEFANIE BRYCE/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

64828 7590 08/06/2013

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

EXAMINER	
CHO, HONG SOL	

ART UNIT	PAPER NUMBER
2467	

NOTIFICATION DATE	DELIVERY MODE
08/06/2013	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

CLAYMCGURK@COX.NET

Advisory Action Before the Filing of an Appeal Brief	**Application No.** 13/531,294	**Applicant(s)** DELANGIS, ERIC M.
	Examiner HONG CHO	**Art Unit** 2467

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 25 July 2013 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

NO NOTICE OF APPEAL FILED

1. ☒ The reply was filed after a final rejection. No Notice of Appeal has been filed. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance;

(2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114 if this is a utility or plant application. Note that RCEs are not permitted in design applications. The reply must be filed within one of the following time periods:

a) ☐ The period for reply expires _____months from the mailing date of the final rejection.

b) ☒ The period for reply expires on: (1) the mailing date of this Advisory Action; or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.

c) ☐ A prior Advisory Action was mailed more than 3 months after the mailing date of the final rejection in response to a first after-final reply filed within 2 months of the mailing date of the final rejection. The current period for reply expires months from the mailing date of *the prior Advisory Action* or SIX MONTHS from the mailing date of the final rejection, whichever is earlier.

Examiner Note: If box 1 is checked, check either box (a), (b) or (c). ONLY CHECK BOX (b) WHEN THIS ADVISORY ACTION IS THE FIRST RESPONSE TO APPLICANT'S FIRST AFTER-FINAL REPLY WHICH WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. ONLY CHECK BOX (c) IN THE LIMITED SITUATION SET FORTH UNDER BOX (c). See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) or (c) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. ☐ The Notice of Appeal was filed on _____. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☒ The proposed amendments filed after a final rejection, but prior to the date of filing a brief, will not be entered because

a) ☒ They raise new issues that would require further consideration and/or search (see NOTE below);

b) ☐ They raise the issue of new matter (see NOTE below);

c) ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or

d) ☐ They present additional claims without canceling a corresponding number of finally rejected claims.

NOTE: *New claims changed the scope of the invention*. (See 37 CFR 1.116 and 41.33(a)).

4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).

5. ☐ Applicant's reply has overcome the following rejection(s): _____.

6. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).

7. ☐ For purposes of appeal, the proposed amendment(s): (a) ☐ will not be entered, or (b) ☐ will be entered, and an explanation of how the new or amended claims would be rejected is provided below or appended.

AFFIDAVIT OR OTHER EVIDENCE

8. ☐ The affidavit or other evidence filed after final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).

9. ☐ The affidavit or other evidence filed after the date of filing the Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing of good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).

10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. ☐ The request for reconsideration has been considered but does NOT place the application in condition for allowance because: _____.

12. ☐ Note the attached Information *Disclosure Statement*(s). (PTO/SB/08) Paper No(s). _____

13. ☐ Other: _____.

STATUS OF CLAIMS

14. The status of the claim(s) is (or will be) as follows:

Claim(s) allowed: .

Claim(s) objected to: .

Claim(s) rejected: .

Claim(s) withdrawn from consideration: .

/HONG CHO/
Primary Examiner, Art Unit 2467

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Mail Stop **AF**
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

DO NOT ENTER: /H.C./

AMENDMENT AFTER FINAL

In response to the final Office Action issued on May 29, 2012 ("Office Action"), please reconsider the above-identified application in light of the following amendments to the claims and accompanying remarks.

MAIL STOP AF

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION

I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: July 25, 2013 By: Clay McGurk
Clay McGurk

RED HAT 1012
RED HAT V. CAS

To: CLAYMCGURK@COX.NET,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

Aug 06, 2013 05:28:41 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	CTAV	08/06/2013	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL
(Submitted Only via EFS-Web)

Application Number	13531294	Filing Date	2012-06-22	Docket Number (if applicable)	ED 1040	Art Unit	2467
First Named Inventor	ERIC DELANGIS			Examiner Name	HONG CHO		

This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.
Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. The Instruction Sheet for this form is located at WWW.USPTO.GOV

SUBMISSION REQUIRED UNDER 37 CFR 1.114

Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).

☒ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.

☒ Consider the arguments in the Appeal Brief or Reply Brief previously filed on 2013-07-25

☐ Other _____

☐ Enclosed

☐ Amendment/Reply

☐ Information Disclosure Statement (IDS)

☐ Affidavit(s)/ Declaration(s)

☐ Other _____

MISCELLANEOUS

☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of months _____ (Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)

☐ Other _____

FEES

☐ **The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.**
The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No _____

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

☒ Patent Practitioner Signature

☐ Applicant Signature

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Signature of Registered U.S. Patent Practitioner			
Signature	/H C McGurk IV, Reg No 34964/	Date (YYYY-MM-DD)	2013-08-27
Name	Harold McGurk	Registration Number	34964

This collection of information is required by 37 CFR 1.114. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Patent Application Fee Transmittal

Application Number:	13531294
Filing Date:	22-Jun-2012
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Filer:	Harold C. McGurk
Attorney Docket Number:	ED 1040

Filed as Small Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

RED HAT 1012
RED HAT V. CAS

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Request for Continued Examination	2801	1	600	600
		Total in USD ($)		**600**

RED HAT 1012
RED HAT V. CAS

Electronic Acknowledgement Receipt

EFS ID:	16695749
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	27-AUG-2013
Filing Date:	22-JUN-2012
Time Stamp:	13:22:22
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$600
RAM confirmation Number	14206
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)

RED HAT 1012
RED HAT V. CAS

1	Request for Continued Examination (RCE)	13531294_RCE_30e_fill.pdf	697835 8f15d781304b4e2a6230cbdaa005a49af7ab3e1d	no	3
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30357 0259fde7a7d4799e7f6e2b39dc9a705134bdeeab	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	728192		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/531,294	Filing Date 06/22/2012	☐ To be Mailed

ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)		
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

		(Column 1)		(Column 2)	(Column 3)		
	08/27/2013	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
AMENDMENT	Total (37 CFR 1.16(i))	* 16	Minus	** 20	= 0	X $40 =	0
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $210 =	0
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	**0**

		(Column 1)		(Column 2)	(Column 3)		
		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
AMENDMENT	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/DEBORAH NASH/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

64828 7590 09/13/2013

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

EXAMINER	
CHO, HONG SOL	

ART UNIT	PAPER NUMBER
2467	

NOTIFICATION DATE	DELIVERY MODE
09/13/2013	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

CLAYMCGURK@COX.NET

Office Action Summary	Application No. 13/531,294	Applicant(s) DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08/27/2013*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *17-37* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *17-37* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some * c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. This office action is in response to the RCE filed on 08/27/13. Claims 1-16 were cancelled. Claims 17-37 are pending in the instant application.

Claim Rejections - 35 USC § 112, First paragraph

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 17-37 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter, which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Re claims 17, 30 and 35, the original specification fails to explain how routing data across the wireless connection increases bandwidth and data throughput at the first device than is achievable over the connection between the first device and the network.

Claims 18-29, 31-34, 36 and 37 depend on claims 17, 30 and 35 are therefore similarly rejected.

RED HAT 1012
RED HAT V. CAS

4. Claims 20-26 and 32-34 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Re claim 20, the original specification fails to explain how to create a data stream from the data aggregated from the wireless connection and the connection at the first location.

Re claim 21, the original specification fails to explain how to create a data file from the data aggregated from the wireless connection and the connection at the first location.

Re claim 22, the original specification fails to explain how to combine and sequence the data received over the wireless connection and the data received over the connection at the first location.

Re claim 23, the original specification fails to explain how to create a data stream from the data combined and sequenced from the wireless connection and the connection at the first location.

Re claim 24, the original specification fails to explain how to create a data file from the data combined and sequenced from the wireless connection and the connection at the first location.

RED HAT 1012
RED HAT V. CAS

Re claim 25, the original specification fails to explain how to create first data stream from the data received over the wireless connection and a second data stream from the data received over the connection at the first location.

Re claim 26, the original specification fails to explain how to create a first data file from the data received over the wireless connection and a second data file from the data received over the connection at the first location.

Re claim 32, the original specification fails to explain how to create a data stream from the data aggregated from the first network and the second network.

Re claim 33, the original specification fails to explain how to create a data file from the data aggregated from the first network and the second network.

Re claim 34, the original specification fails to explain how to create a first data file from the data received from the first network and a second data file from the data received from the second network.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

RED HAT 1012
RED HAT V. CAS

6. Claims 17-20, 22, 23, 25, 27-32 and 35-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vehvilainen (US 7426316) in view of Souisse et al (US 20020102987, "Souisse").

Re claims 17, 30 and 35-37, Vehvilainen discloses establishing a wireless connection between a first device and a second device, the first device being connected at a first location to a network, the second device being connected at a second location to the network (figure 1; column 5, lines 11-16), the second location being geographically separate from the first location, the first device and the second device being configured for accessing the network (column 4, lines 29-33), but fails to disclose routing data across the wireless connection thereby increasing bandwidth and data throughput at the first device than is achievable over the connection between the first device and the network. However, Souisse discloses increasing utilization of the wireless communications to provide more bandwidth and throughput (paragraph [0012]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Vehvilainen with the teaching of Souisse for the benefit of providing efficient utilization of network resources.

Re claim 18, Vehvilainen discloses the first location and the second location are physically isolated from each other (figure 1).

Re claims 19, 20, 22, 23, 25, 31 and 32, Vehvilainen discloses all of the limitations of the base claim, but fails to disclose aggregating the data received over the wireless connection and the data received over the connection at the first location.

RED HAT 1012
RED HAT V. CAS

Souisse discloses combining fragmented messages into an original message (paragraph [0038]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Vehvilainen with the teaching of Souisse for the benefit of providing efficient network communication by combining data streams in sequence.

Re claims 27-29, Vehvilainen discloses the wireless connection is established between a first wireless router at the first device and a second wireless router at the second device, wherein the network is one or a combination of a cable network, a telecommunication network, an optical network and a communications network and the wireless connection is a cellular, satellite, Wi-Fi, WiMax or Bluetooth connection (column 5, lines 11-16).

Claims 21, 24, 26, 33 and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vehvilainen in view of Souisse and Miloushev et al (US 7562110, "Miloushev").

Re claims 21, 24, 26, 33 and 34, Vehvilainen discloses all of the limitations of the base claim, but fails to disclose creating a data file from the data combined and sequenced from the wireless connection and the connection at the first location. However, Miloushev discloses creating a data file from the data combined and sequenced from the wireless connection and the connection at the first location (column 21, lines 55-60). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Vehvilainen with the teaching of

RED HAT 1012
RED HAT V. CAS

Miloushev for the benefit of providing efficient management of combined files by referring to a single file data.

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hong Cho whose telephone number is 571-272-3087. The examiner can normally be reached on Mon-Fri during 7 am to 4 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hassan Phillips can be reached on 571-272-3940. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Hong Cho/

Primary Examiner, Art Unit 2467

RED HAT 1012
RED HAT V. CAS

Notice of References Cited	Application/Control No. 13/531,294	Applicant(s)/Patent Under Reexamination DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-7,426,316	09-2008	Vehvilainen, Markku	382/275
*	B	US-7,562,110	07-2009	Miloushev et al.	709/201
*	C	US-2002/0102987	08-2002	Souisse et al.	455/454
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	11/26/2012	05/21/2013	09/05/2013						
	1	✓	✓	-						
	2	✓	✓	-						
	3	✓	✓	-						
	4	✓	✓	-						
	5	✓	✓	-						
	6	✓	✓	-						
	7	✓	✓	-						
	8	✓	✓	-						
	9		✓	-						
	10		✓	-						
	11		✓	-						
	12		✓	-						
	13		✓	-						
	14		✓	-						
	15		✓	-						
	16		✓	-						
	17			✓						
	18			✓						
	19			✓						
	20			✓						
	21			✓						
	22			✓						
	23			✓						
	24			✓						
	25			✓						
	26			✓						
	27			✓						
	28			✓						
	29			✓						
	30			✓						
	31			✓						
	32			✓						
	33			✓						
	34			✓						
	35			✓						
	36			✓						

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	Art Unit
	HONG CHO	2467

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	11/26/2012	05/21/2013	09/05/2013						
	37			✓						

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	9	((ERIC) near2 (DELANGIS)).INV.	US-PGPUB; USPAT; USOCR	OR	OFF	2013/09/05 09:49
L17	10	L16 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/09/05 10:33
L16	83	(aggregat$3 or combin$3 or multiplex$3) near received near (packet or frame or cell or data or message or inforamtion) with sequenc$3	US-PGPUB; USPAT	OR	OFF	2013/09/05 10:33
L20	30	(combin$5 or aggregat$3 or multiplex$3) with (receiv$3 or transmit$4) with wireless near connection with internet	US-PGPUB; USPAT	OR	OFF	2013/09/05 10:48
L21	1	"7426316".pn.	US-PGPUB; USPAT	OR	OFF	2013/09/05 10:50
L28	13	L26 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/09/05 10:58
L26	160	(combin$5 or aggregat$3 or multiplex$3) with received with (data or flow$1 or stream$1)with wireless with internet	US-PGPUB; USPAT	OR	OFF	2013/09/05 10:58
L31	237	(combin$5 or aggregat$3 or multiplex$3) with (receiv$3 or transmit$4) near (packet$1 or frame$1 or cell$1 or data or message or inforamtion) with wireless with internet	US-PGPUB; USPAT	OR	OFF	2013/09/05 11:15
L32	25	L31 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/09/05 11:16
L40	144	(router or gateway) with (multiplex$3 or combin$5 or aggregat$3) with (packet$1 or frame$1 or cell$1 or data or message or information) near (transmitted or received) with wireless$2	US-PGPUB; USPAT	OR	OFF	2013/09/05 11:31
L43	224	L42 and (multiplex$3 or aggregat$3 or combin$3) near (traffic or data or message$1 or inforamtion or packet$1 or frame$1) with (wireless or mobile) adj (devices or station$1)	US-PGPUB; USPAT	OR	OFF	2013/09/05 11:35
L42	42135	370/230-235,278,282,328,338,468,477.ccls.	US-PGPUB; USPAT	OR	OFF	2013/09/05 11:35
L51	15	L50 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/09/05 11:52
L50	135	(multiplex$3 or aggregat$3 or combin$3) with wireless$2 near received near (stream$1 or flow$1 or traffic or data or message$1 or inforamtion or packet$1 or frame$1)	US-PGPUB; USPAT	OR	OFF	2013/09/05 11:52

L52	1	13/531294	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2013/09/05 11:55
L55	5	L53 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/09/05 11:59
L53	78	(ms or mobile adj (station$1 or device$1)) with wireless$3 with establish$4 near connection with (ap or internet or ip)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2013/09/05 11:59
L57	166	L56 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/09/05 12:00
L56	1515	wireless$3 with establish$4 near connection with (ap or internet or ip)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2013/09/05 12:00
L60	41	L59 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/09/05 12:48
L59	55	increas$3 near (bandwidth or throughput) with wireless near connection	US-PGPUB; USPAT	OR	OFF	2013/09/05 12:48
L64	29	L63 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/09/05 12:51
L63	252	increas$3 with (bandwidth with throughput) with wireless	US-PGPUB; USPAT	OR	OFF	2013/09/05 12:51
L70	51	L69 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/09/05 13:25
L69	267	creat$3 near file with data near (flow$1 or stream$1)	US-PGPUB; USPAT	OR	OFF	2013/09/05 13:25
L71	1	"7562110".pn.	US-PGPUB; USPAT	OR	OFF	2013/09/05 13:28
L72	1	13/531294	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2013/09/05 14:06

9/5/2013 2:44:34 PM
C:\Users\hcho\Documents\EAST\Workspaces\13531294.wsp

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	Art Unit
	HONG CHO	2467

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
370	230-235,278,282,328,338,468,477	11/26/2012	hc

SEARCH NOTES		
Search Notes	Date	Examiner
EAST text search	11/26/2012	hc
updated search	5/21/2013	hc
updated search	9/5/2013	hc

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

To: CLAYMCGURK@COX.NET,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

Sep 13, 2013 05:22:28 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	CTNF	09/13/2013	ED 1040
	892	09/13/2013	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Mail Stop **Amendment**
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

AMENDMENT

In response to the Office Action issued on September 13, 2013 ("Office Action"), please reconsider the above-identified application in light of the following amendments to the claims and accompanying remarks.

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION

I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: December 12, 2013

By: Clay McGurk
Clay McGurk

RED HAT 1012
RED HAT V. CAS

THE CLAIMS

This listing of claims will replace all prior versions and listings of the claims in the application:

1-16. (Canceled).

17. (Currently Amended) A method for increasing bandwidth and data throughput comprising the steps of:

establishing a wireless connection between a first device and a second device, the first device being connected at a first location to a network via a landline, the second device being connected at a second location to the network, the second location being geographically separate from the first location, the first device and the second device being configured for accessing the network; and

~~routing data across the wireless connection thereby increasing bandwidth and data throughput at the first device than is achievable over the connection between the first device and the network~~ concurrently receiving at the first location first data from the network and second data over the wireless connection, the second data being received at the second location from the network.

18. (Previously Presented) The method of claim 17, wherein the first location and the second location are physically isolated from each other.

19. (Currently Amended) The method of claim 17, further comprising the step of aggregating the second data received over the wireless connection and the first data received over the connection at the first location.

20. (Currently Amended) The method of claim ~~19~~ 17, ~~further comprising the step of creating a data stream from the data aggregated from the wireless connection and the connection at the first location~~ wherein a connection at the first location and a connection at the second location are network connection points.

RED HAT 1012
RED HAT V. CAS

21. (Currently Amended) The method of claim ~~19~~ 17, ~~further comprising the step of creating a data file from the data aggregated from the wireless connection and the connection at the first location~~ wherein the first device and the second device are consumer networking devices.

22. (Currently Amended) The method of claim 17, further comprising the ~~step~~ steps of; ~~combining and sequencing the data received over the wireless connection and the data received over the connection at the first location~~

establishing a second wireless connection between the first device and a third device, the third device being connected at a third location to the network, the third location being geographically different than the first location and the second location; and

concurrently receiving third data over the second wireless connection with the receipt of the first data at the first location and the second data over the wireless connection.

23. (Currently Amended) The method of claim 22, further comprising the step of ~~creating a data stream from the data combined and sequenced from the wireless connection and the connection at the first location~~ aggregating the first data, the second data and the third data.

24. (Canceled).

25. (Currently Amended) The method of claim 17, ~~further comprising the step of creating a first data stream from the data received over the wireless connection and a second data stream from the data received over the connection at the first location~~ wherein the first data and the second data are received as data packets.

26. (Currently Amended) The method of claim 17, ~~further comprising the step of creating a first data file from the data received over the wireless connection and a second data file from the data received over the connection at the first location~~ wherein the network is the Internet.

RED HAT 1012
RED HAT V. CAS

27. (Previously Presented) The method of claim 17, wherein the wireless connection is established between a first wireless router at the first device and a second wireless router at the second device.

28. (Previously Presented) The method of claim 17, wherein the network is one or a combination of a cable network, a telecommunication network, an optical network and a communications network.

29. (Currently Amended) The method of claim 17, wherein the wireless connection is a cellular connection, a satellite connection, a Wi-Fi connection, a WiMax connection or a Bluetooth connection.

30-34. (Canceled).

35. (Currently Amended) A device comprising:

an interface ~~for establishing~~ that establishes a landline network connection at a first location to a first network and receives first data packets over the network connection;

a wireless interface ~~for establishing~~ that establishes a wireless connection to a remote communications device, the remote communications device being connected at a second location to ~~the~~ a second network, the second location being geographically separate from the first location; and

a processor that receives from the interface first data packets from the first network and concurrently receives over the wireless interface second data packets from the second network, the second data packets being received at the second location from the second network ~~requests the remote communications device to assist in routing data over the wireless connection, receives the data over the wireless connection thereby increasing communications bandwidth and data throughput than is achievable over the network connection~~.

36. (Previously Presented) The device of claim 35, wherein the device is one of a modem, cellular phone, a router, a computer, hotspot device and a tablet.

37. (Currently Amended). The device of claim 35, wherein the ~~network connection is a wireless connection~~ the device is part of a home-based network system.

38. (New) The device of claim 35, wherein the first network and the second network are the same network.

39. (New) A method for providing broadband communications services, comprising:

receiving first data packets over a landline connection;

receiving second data packets over a wireless connection, the first data packets being concurrently received with the second data packets; and

aggregating the first data packets with the second data packets.

40. (New) The method of claim 39, wherein the first data packets are received at a home-based network system, and wherein the second packets are received over the wireless connection at the home-based network system.

41. (New) The method of claim 39, wherein the first data packets are received by a consumer-based network system, and wherein the second packets are received over the wireless connection at the consumer-based network system.

42. (New) The method of claim 39, further comprising:

receiving data packets over a plurality of wireless connections, the data packets being concurrently received with the first data packets; and

aggregating the first data packets with the data packets.

REMARKS

Status of the Claims

Claims 17-37 were pending in this application before this Amendment. In this Amendment, claims 24 and 30-34 are canceled without prejudice or disclaimer. Claims 17, 19-23, 25-26, 29, 35 and 37 are amended herein. Claims 39-42 are new claims and are added in this Amendment. No new matter has been added. Thus, claims 17-23, 25-29, 35-42 are currently pending in this application pursuant to this Amendment.

35 U.S.C. § 112 Rejections

On pages 2-4 of the Office Action, claims 17-37 were rejected under 35 U.S.C. §112, first paragraph. All the claims which were rejected have been either amended or canceled. The claims have been amended to advance prosecution without addressing the alleged merits of the §112 rejection. The applicant believes there is ample support in the specification for what was previously claimed and specifically reserves the right to use such deleted language in the future. It is respectfully requested that the rejection of claims 17-37 be reconsidered and withdrawn.

35 U.S.C. § 103 Rejections

On pages 5-6 of the Office Action, claims 17-20, 22-23, 25, 27-32 and 35-37 were rejected under 35 U.S.C. §103(a) as being unpatentable over Vehvilainen (U.S. Patent 7,426,316) ("Vehvilainen") and Souisse *et al.* (U.S. Application Publication 2002-0102987) ("Souisse").

Vehvilainen is poor prior art because this reference describes a method and apparatus for improving image quality, which has nothing to do with the claimed invention. Vehvilainen does not describe or suggest concurrently receiving data as required in amended claims 17 and 35, or aggregating data as in new claim 39. Though image phone 102 and personal computer 104 may be connected to a network 103 or each other, Vehvilainen fails to describe or suggest concurrently receiving data. (Vehvilainen, Fig. 1; 4:29-33). Vehvilainen is a point-to-point connection between the phone 102 and computer 104, either via communication network 103 or via a Bluetooth connection. (Vehvilainen, 5:11-16). Data is sent between the phone 102 and computer 104 over one of the connection, but not both concurrently.

RED HAT 1012
RED HAT V. CAS

Souisse does not cure the deficiency of Vehvilainen because it is concerned with fragmenting a single message and the sending the message fragments over multiple independent networks to a single receiving unit. (Souisse, Abstract; Figs. 5, 6). Souisse was used for rejecting the prior claim language of claims 17, 30 and 35, rather than the amended language currently used in these claims which requires a landline connection, something missing in Souisse. Therefore a combination a Souisse and Vehvilainen does not result in the claimed invention because both fail to describe or suggest, either individually or in combination, each and all the elements required in claims 17 and 35. It is respectfully requested that the rejection of claims 17-20, 22-23, 25, 27-32 and 35-37 under 35 U.S.C. §103(a) based on Vehvilainen and Souisse be reconsidered and withdrawn.

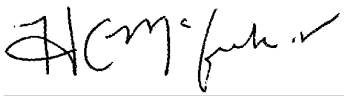
On page 6-7 of the Office Action, claims 21, 24, 26, 33 and 34 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Vehvilainen in view of Souisse and Miloushev *et al.* (U.S. Patent 7,562,110). This rejection was based on claim language that was deleted and replaced with different claim language having different subject matter. It is therefore respectfully requested that the rejection of claims 21, 24, 26, 33 and 34 be reconsidered and withdrawn.

Conclusion

All rejections in the Office Action have been properly traversed. It is respectfully requested that all outstanding rejections be reconsidered and withdrawn. It is believed that a full and complete reply has been made in response to the Office Action, and, that as such, the present patent application is in a condition for allowance. The Examiner is always welcome to telephone the attorney of record at the telephone number provided below if such telephone discussion would help expedite the prosecution of the patent application.

Prompt and favorable consideration of this Amendment is respectfully requested.

Respectfully Submitted,



Date: December 12, 2013

H C McGurk IV, Reg. No. 34,964
Attorney for Applicant

RED HAT 1012
RED HAT V. CAS

The Law Office of Clay McGurk
PO Box 1488
Orange, CA 92706
Phone: **949.395.3942**
Email: claymcgurk@gmail.com

Electronic Acknowledgement Receipt

EFS ID:	17648282
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	12-DEC-2013
Filing Date:	22-JUN-2012
Time Stamp:	19:17:29
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Amendment/Req. Reconsideration-After Non-Final Reject	13531294_Amendment_12-12-13.pdf	97937 34a13fe792b64432709609eb0da7a12f455b79f5	no	8

Warnings:	
Information:	

RED HAT 1012
RED HAT V. CAS

Total Files Size (in bytes):	97937

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/531,294	Filing Date 06/22/2012	☐ To be Mailed

ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT **12/12/2013**

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 20	Minus	** 21	= 0	X $40 =	0
Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $210 =	0
☐ Application Size Fee (37 CFR 1.16(s))						
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
					TOTAL ADD'L FEE	0

AMENDMENT

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))						
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
					TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/ROSALIND BALL/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

64828 7590 12/30/2013

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

EXAMINER	
CHO, HONG SOL	

ART UNIT	PAPER NUMBER
2467	

NOTIFICATION DATE	DELIVERY MODE
12/30/2013	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

CLAYMCGURK@GMAIL.COM

Office Action Summary	Application No. 13/531,294	Applicant(s) DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12/12/13*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *17-23,25-29 and 35-42* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *17-23, 25-29 and 35-42* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some * c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. This office action is in response to the amendment filed on 12/12/13. Claims 17-23, 25-29 and 35-42 are pending in the instant application.

Claim Rejections - 35 USC § 112, First paragraph

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 17-23, 25-29 and 35-42 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter, which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Re claim 17, the original specification fails to disclose concurrently receiving at the first location first data from the network and second data over the wireless connection, the second data being received at the second location from the network.

Re claim 22, the original specification fails to disclose establishing a second wireless connection between the first device and a third device, the third device being

RED HAT 1012
RED HAT V. CAS

connected at a third location to the network, the third location being geographically different than the first location and the second location; and concurrently receiving third data over the second wireless connection with the receipt of the first data at the first location and the second data over the wireless connection.

Re claim 35, the original specification fails to disclose a processor that receives from the interface first data packets from the first network and concurrently receives over the wireless interface second data packets from the second network, the second data packets being received at the second location from the second network.

Re claim 39, the original specification fails to disclose receiving second data packets over a wireless connection, the first data packets being concurrently received with the second data packets

Re claim 42, the original specification fails to disclose receiving data packets over a plurality of wireless connections, the data packets being concurrently received with the first data packets; and aggregating the first data packets with the data packets.

Claims 18-23, 25-29, 36-38 and 40-42 depend on claims 17, 35 and 39 are therefore similarly rejected.

Claim Rejections - 35 USC § 112 Second Paragraph

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

RED HAT 1012
RED HAT V. CAS

5. Claims 40 and 41 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Re claims 40 and 41, it is not clear what home-based/consumer-based network system is.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 17-23, 25-29 and 35-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vehvilainen (US 7426316) in view of Bi (US 7113173).

Re claims 17-21, 25, 26, 35-40 and 42, Vehvilainen discloses establishing a wireless connection between a first device and a second device, being a consumer device, the first device being connected at a first location to a network, the second device being connected at a second location to the network (figure 1; column 5, lines 11-16), the second location being geographically separate from the first location, the first device and the second device being configured for accessing the network (column 4, lines 29-33),

RED HAT 1012
RED HAT V. CAS

but fails to disclose concurrently receiving data from networks by landline and wireless connection. However, Bi discloses multiplexing received packets from wired and wireless connections (figures 44 and 46, column 47, lines 23-35). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Vehvilainen with the teaching of Bi for the benefit of providing efficient utilization of network resources by multiplexing data packets.

Re claim 18, Vehvilainen discloses the first location and the second location are physically isolated from each other (figure 1).

Re claims 22 and 23, Vehvilainen discloses establishing a second wireless connection between the first device and a third device and concurrently receiving third data over the second wireless connection with the receipt of the first data at the first location and the second data over the wireless connection (column 4, lines 29-33). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Vehvilainen with the teaching of Bi for the benefit of providing efficient utilization of network resources by multiplexing data packets.

Re claims 27-29, Vehvilainen discloses the wireless connection is established between a first wireless router at the first device and a second wireless router at the second device, wherein the network is one or a combination of a cable network, a telecommunication network, an optical network and a communications network and the wireless connection is a cellular, satellite, Wi-Fi, WiMax or Bluetooth connection (column 5, lines 11-16).

RED HAT 1012
RED HAT V. CAS

Response to Arguments

8. Applicant's arguments with respect to claims have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hong Cho whose telephone number is 571-272-3087. The examiner can normally be reached on Mon-Fri during 7 am to 4 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hassan Phillips can be reached on 571-272-3940. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

RED HAT 1012
RED HAT V. CAS

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Hong Cho/

Primary Examiner, Art Unit 2467

RED HAT 1012
RED HAT V. CAS

Notice of References Cited	Application/Control No. 13/531,294	Applicant(s)/Patent Under Reexamination DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-7,113,173	09-2006	Bi et al.	345/169
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	Art Unit
	HONG CHO	2467

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
370	230-235,278,282,328,338,468,477	11/26/2012	hc

SEARCH NOTES		
Search Notes	Date	Examiner
EAST text search	11/26/2012	hc
updated search	5/21/2013	hc
updated search	9/5/2013	hc
updated search	12/23/2013	hc

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L2	16	L1 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/12/23 11:59
L1	141	(multiplex$3 or aggregat$3 or combin$3) with wireless$2 near received near (stream$1 or flow$1 or traffic or data or message$1 or inforamtion or packet$1 or frame$1)	US-PGPUB; USPAT	OR	OFF	2013/12/23 11:59
L14	568	concurrently near receiv$3 near (data or packet$1 or frame$1)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2013/12/23 12:43
L15	111	L14 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/12/23 12:45
L17	75	L14 and @rlad<"20021015" and wireless	US-PGPUB; USPAT	OR	OFF	2013/12/23 12:46
L28	29531	(multiplex$3 or aggregat$3 or combin$3) with received near (data or message$1 or packet$1 or frame$1)	US-PGPUB; USPAT	OR	OFF	2013/12/23 13:12
L29	5211	L28 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/12/23 13:13
L40	26	L39 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2013/12/23 13:21
L39	93	(multiplex$3 or aggregat$3 or combin$3) with (data or packet$1 or frame$1) near wired with wireless$2	US-PGPUB; USPAT	OR	OFF	2013/12/23 13:21

12/23/2013 2:42:33 PM
C:\Users\hcho\Documents\EAST\Workspaces\13531294.wsp

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	11/26/2012	05/21/2013	09/05/2013	12/23/2013					
	1	✓	✓	-	-					
	2	✓	✓	-	-					
	3	✓	✓	-	-					
	4	✓	✓	-	-					
	5	✓	✓	-	-					
	6	✓	✓	-	-					
	7	✓	✓	-	-					
	8	✓	✓	-	-					
	9		✓	-	-					
	10		✓	-	-					
	11		✓	-	-					
	12		✓	-	-					
	13		✓	-	-					
	14		✓	-	-					
	15		✓	-	-					
	16		✓	-	-					
	17			✓	✓					
	18			✓	✓					
	19			✓	✓					
	20			✓	✓					
	21			✓	✓					
	22			✓	✓					
	23			✓	✓					
	24			✓	-					
	25			✓	✓					
	26			✓	✓					
	27			✓	✓					
	28			✓	✓					
	29			✓	✓					
	30			✓	-					
	31			✓	-					
	32			✓	-					
	33			✓	-					
	34			✓	-					
	35			✓	✓					
	36			✓	✓					

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	11/26/2012	05/21/2013	09/05/2013	12/23/2013					
	37			✓	✓					
	38				✓					
	39				✓					
	40				✓					
	41				✓					
	42				✓					

To: CLAYMCGURK@GMAIL.COM,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

Dec 30, 2013 05:29:50 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	CTFR	12/30/2013	ED 1040
	892	12/30/2013	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE
COMMISSIONER FOR PATENTS
P.O.BOX 1450
ALEXANDRIA VA 22313-1451

PRESORTED
FIRST-CLASS MAIL
U.S. POSTAGE PAID
POSTEDIGITAL
NNNNN

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

Courtesy Reminder for
Application Serial No: 13/531,294

Attorney Docket No: ED 1040
Customer Number: 64828
Date of Electronic Notification: 12/30/2013

This is a courtesy reminder that new correspondence is available for this application. If you have not done so already, please review the correspondence. The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

An email notification regarding the correspondence was sent to the following email address(es) associated with your customer number:
CLAYMCGURK@GMAIL.COM

To view your correspondence online or update your email addresses, please visit us anytime at **https://sportal.uspto.gov/secure/myportal/privatepair.** If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov or call 1-866-217-9197.

AMENDMENT UNDER 37 C.F.R. §1.116
EXPEDITED PROCEDURE
Technology Center 2400

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Mail Stop **AF**
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

AMENDMENT UNDER 37 C.F.R. §1.116

In response to the final Office Action issued on December 30, 2013 ("Office Action"), please reconsider the above-identified application in light of the following amendments to the claims and the accompanying remarks.

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION

I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: March 11, 2014

By: Clay McGurk
Clay McGurk

RED HAT 1012
RED HAT V. CAS

THE CLAIMS

This listing of claims will replace all prior versions and listings of the claims in the application:

1-16. (Canceled).

17. (Currently Amended) A method for increasing bandwidth and data throughput comprising the steps of:

establishing a wireless connection between a first device and a second device, the first device being connected at a first location to a network via a landline, the second device being connected at a second location to the network, the second location being geographically separate from the first location, the first device and the second device being configured for accessing the network; and

concurrently receiving at the first ~~location~~ device first data from the network at the first location and second data over the wireless connection from the second device, the second data being received at the second location from the network.

18. (Previously Presented) The method of claim 17, wherein the first location and the second location are physically isolated from each other.

19. (Previously Presented) The method of claim 17, further comprising the step of aggregating the second data received over the wireless connection and the first data received over the connection at the first location.

20. (Previously Presented) The method of claim 17, wherein a connection at the first location and a connection at the second location are network connection points.

21. (Previously Presented) The method of claim 17, wherein the first device and the second device are consumer networking devices.

RED HAT 1012
RED HAT V. CAS

22. (Previously Presented) The method of claim 17, further comprising the steps of; establishing a second wireless connection between the first device and a third device, the third device being connected at a third location to the network, the third location being geographically different than the first location and the second location; and

concurrently receiving third data over the second wireless connection with the receipt of the first data at the first location and the second data over the wireless connection.

23. (Previously Presented) The method of claim 22, further comprising the step of aggregating the first data, the second data and the third data.

24. (Canceled).

25. (Previously Presented) The method of claim 17, wherein the first data and the second data are received as data packets.

26. (Previously Presented) The method of claim 17, wherein the network is the Internet.

27. (Previously Presented) The method of claim 17, wherein the wireless connection is established between a first wireless router at the first device and a second wireless router at the second device.

28. (Previously Presented) The method of claim 17, wherein the network is one or a combination of a cable network, a telecommunication network, an optical network and a communications network.

29. (Previously Presented) The method of claim 17, wherein the wireless connection is a cellular connection, a satellite connection, a Wi-Fi connection, a WiMax connection or a Bluetooth connection.

30-34. (Canceled).

35. (Previously Presented) A device comprising:

an interface that establishes a landline network connection at a first location to a first network and receives first data packets over the network connection;

a wireless interface that establishes a wireless connection to a remote communications device, the remote communications device being connected at a second location to a second network, the second location being geographically separate from the first location; and

a processor that receives from the interface first data packets from the first network and concurrently receives over the wireless interface second data packets from the second network, the second data packets being received at the second location from the second network.

36. (Previously Presented) The device of claim 35, wherein the device is one of a modem, cellular phone, a router, a computer, hotspot device and a tablet.

37. (Currently Amended) The device of claim 35, wherein the ~~the~~ device is part of a home-based network system.

38. (Previously Presented) The device of claim 35, wherein the first network and the second network are the same network.

39. (Previously Presented) A method for providing broadband communications services, comprising:

receiving first data packets over a landline connection;

receiving second data packets over a wireless connection, the first data packets being concurrently received with the second data packets; and

aggregating the first data packets with the second data packets.

40. (Previously Presented) The method of claim 39, wherein the first data packets are received at a home-based network system, and wherein the second packets are received over the wireless connection at the home-based network system.

RED HAT 1012
RED HAT V. CAS

41. (Previously Presented) The method of claim 39, wherein the first data packets are received by a consumer-based network system, and wherein the second packets are received over the wireless connection at the consumer-based network system.

42. (Previously Presented) The method of claim 39, further comprising:

receiving data packets over a plurality of wireless connections, the data packets being concurrently received with the first data packets; and

aggregating the first data packets with the data packets.

RED HAT 1012
RED HAT V. CAS

REMARKS

Status of the Claims

Claims 17-37 were pending in this application before this Amendment.

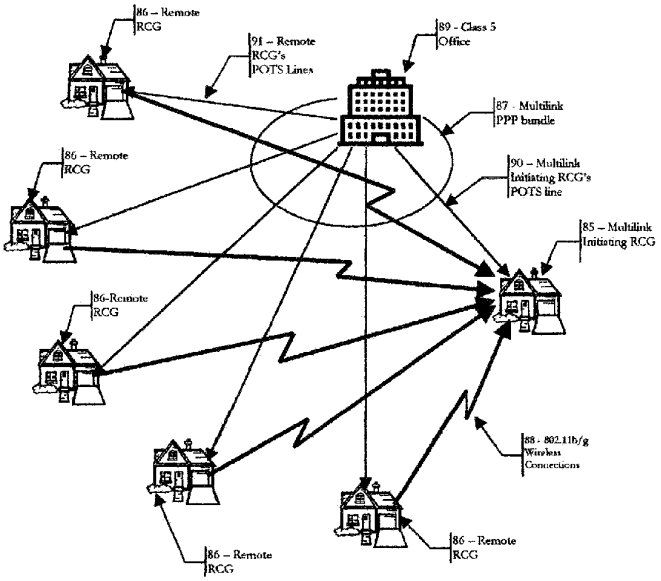
Claims 17 and 37 are amended according to 37 C.F.R. §1.116. Claim 17 is being amended to further clarify that the first device is receiving the first data and the second data. The changes to claim 17 were made to make this method claim more similar to what is being claimed in device claim 35. Claim 37 is being amended to delete a second occurrence of "the". No new matter has been added.

Thus, claims 17-23, 25-29, 35-42 are currently pending in this application pursuant to this Amendment.

35 U.S.C. § 112 (first paragraph) Rejection

On pages 2-3 of the Office Action, claims 17-23, 25-29 and 35-42 were rejected under 35 U.S.C. §112, first paragraph. Issues of "concurrently receiving" are being raised in regards to claims 17, 22, 35, 39 and 42. "Concurrently receiving" is one the central purposes of all embodiments of the present invention - which is to receive data over many different links at the same time. This is clearly shown in Figure 7 of the present application, reproduced below:

FIG.7



RED HAT 1012
RED HAT V. CAS

The multilink initiating RCG 85 is receiving first data over POTS line 90. The five RCG's shown in Fig. 7 are each receiving their data from the network via each of their own POTS line 91. Multilink initiating RCG 85 receives first data over POTS line 90 and concurrently receives data from remote RCGs 86 via wireless connections 88. In one embodiment of the present invention, this is referred to as a "Multilink PPP Connection" which is explained in detail in paragraphs 72-80, among other places. It is inherent in a "multilink PPP connection" that the multilink initiating RCG 85 receives data concurrently thereby fulfilling one of the main purposes of the invention which is to increase bandwidth for a particular location. Otherwise, if data is not being concurrently received, there is no increase in bandwidth and defeats one of the primary purposes of the present invention.

The present application incorporated by reference both the parent and grand-parent applications. The grand-parent application, serial no. 10/686,375, was published in U.S. Published Application 2005-0078690. In that original filing of the grand-parent patent application, there existed claim 3 that related to concurrent reception of data. Claim 3 of the original grand-parent application is reproduced below:

> 3. An RCG of claims 1 and 2 that dynamically allocates separate and physically diverse POTS lines or wireless connections into a multilink group capable of aggregating the combined bandwidth of a plurality of said separate physically diverse POTS lines or wireless connections and providing said aggregate bandwidth to the benefit of a single, or a plurality of RCG device for the concurrent and high speed transmission of large or multiple files; and that can utilize one or more of the separate and physically diverse POTS lines or wireless connections that are physically connected to other remote RCG devices as stand alone connections that are not grouped in a multilink configuration to the benefit of a single or a plurality of RCG devices for the concurrent high speed transmission of large or multiple files.

Fig. 7 undoubtedly shows a device that is able to connect to multiple remote devices and can concurrently receive data from each of the remote devices. Therefore, the specification clearly supports what is being claimed in claims 17, 22, 35, 39 and 42. It is respectfully requested that the rejection of claims 17-23, 25-29 and 35-42 under 35 U.S.C. §112, first paragraph, be reconsidered and withdrawn.

35 U.S.C. § 112 (second paragraph) Rejections

On pages 4-5 of the Office Action, claims 40 and 41 were rejected under 35 U.S.C. §112, second paragraph as being indefinite due to issues about what is meant by a home-based network or a consumer-based network. In paragraphs 9-10 and 12 of the present application (among others), it describes in one embodiment how the device (e.g., RCG) is simply plugged into the existing telephone line at a residence or small business. The phrases "home-based network" and "consumer-based network" should be interpreted using a plain or ordinary meaning to those of ordinary skill in the art in light of what is described in the specification. Therefore, it is respectfully requested that the rejection of claims 40 and 41 under 35 U.S.C. §112, second paragraph, be reconsidered and withdrawn.

35 U.S.C. § 103 Rejections

On pages 5-6 of the Office Action, claims 17-23, 25-29 and 35-42 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent 7,426,316 to Vehvilainen ("Vehvilainen") in view of U.S. Patent 7,113,173 to Bi *et al.* ("Bi"). Basically, Vehvilainen was relied for teaching the first paragraph of claim 17 by citing Fig. 1, col. 5, lines 11-16 and col. 4, lines 29-33. Bi was relied for teaching the second paragraph of claim 17 by citing Figs. 44 and 46, and col. 47, lines 23-35. It will be shown that the Vehvilainen and Bi in combination do not teach all the limitations of claim 17, 35 and 39.

Vehvilainen teaches four ways to transfer images or photographs which are taken on a phone 102 to a computer 104. (Vehvilainen, 5:11-16; 4:29-33). Vehvilainen describes four ways that images are transferred from the phone 102 to the computer 104: (1) through a communication network 103 (e.g., the Internet); or (2) a Bluetooth connection; or (3) a similar wireless connection; or (4) a cable connecting the devices. *Id.* Notably, the citations in Vehvilainen use the word "or" and never the word "and" when describing the different ways of transferring the images. By using the word "or", Vehvilainen only describes that images are transferred only over one of these four ways from the phone 102 to the computer 104, or from another phone to the phone 102. Vehvilainen never describes or suggests using two or more of these transfer methods concurrently to transfer images from the phone 102 to the computer 104. Data is sent between the phone 102 and the computer 104 over one of the connections, but not both concurrently.

Vehvilainen is using what is known to those of ordinary skill in the art as a point-to-point connection. A point-to-point connection described in Vehvilainen is the one of the problems that the present application was trying to solve. The present invention provides advantages of transferring large amounts of information or multiple files over multiple lines at the same time.

There is no description or suggestion that the phone 102 or the computer 104 of Vehvilainen is able to support sending images concurrently over different connections. However, even if Vehvilainen did suggest such, it would still fall short of what is recited in claims 17, 35 and 39. These claims require that first data is being received from the network at the first device (e.g., computer 104), ***but also*** that second data is being received by a second device (e.g., phone 102) from the network and is also transmitted wirelessly to the first device (computer 104) concurrently with the receipt of the first data. In Vehvilainen, image data ***originates*** at the phone 102, **not** from the network 103 as required in the claims. Images in Vehvilainen are transferred from the phone 102 across the network 103 to the computer 104, rather than data being transferred from the network 103 to the phone 102 then to the computer 104. Therefore Vehvilainen is missing a limitation of the claim 17 in that the phone 102 would be receiving data **from** the network and then transferring that data to computer 104.

Vehvilainen is poor prior art because this reference describes a method and apparatus for ***improving image quality***, which has nothing to do with the claimed invention. The methods described in Vehvilainen produce image correcting information to decrease errors in the image produced by a camera module. (Vehvilainen, Abstract). The embodiments of the claimed invention concern ways of transferring data across multiple connections at the same time, rather than ways for improving image quality such as described in Vehvilainen. Therefore, Vehvilainen is a poor reference, unrelated to what is being solved in the present application, and basically is from a different field that is non-analogous art.

Bi does not cure the deficiency of Vehvilainen because Bi is nothing more than describing a router in the portions cited in the Office Action. (Bi, 47:23-35). In Bi, Fig. 46 merely illustrates a software structure of a network system that enables a wireless device to interface with a wired LAN system such as shown in Fig. 44. That is all that Fig. 44 shows – a wireless device 101 being able to connect to a host computer system. In other words, Bi shows in Figs. 44 and 46 a way for a router to communicate with a LAN and a wireless device. The multiplexing of data from the wired LAN card 1352 and the wireless LAN card 1360 is for

transmitting combined data to a network such as the Internet. *Id.* Bi simply allows a wireless device 101 to communicate with the Internet or a LAN system. Bi never describes or suggests a way for the wireless device 101 to transmit data it receives from the network to another device as required in claims 17, 35 and 39.

A combination of Vehvilainen and Bi only results in a system where it makes possible for the phone 102 and the computer 104 to communicate with the network 103. Bi simply is describing a way that is not described in Vehvilainen for ***how*** the phone 102 connects to the network 103, or the computer 104 connects to the network 103. Therefore a combination Vehvilainen and Bi does not result in the claimed invention because both fail to describe or suggest, either individually or in combination, each and all the elements required in claims 17, 35 and 39. It is respectfully requested that the rejection of claims 17-23, 25-29 and 35-42 under 35 U.S.C. §103(a) based on Vehvilainen and Bi be reconsidered and withdrawn.

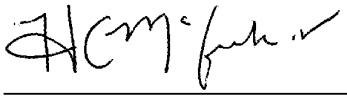
Conclusion

All rejections in the Office Action have been properly traversed. It is respectfully requested that all outstanding rejections be reconsidered and withdrawn. It is believed that a full and complete reply has been made in response to the Office Action, and, that as such, the present patent application is in a condition for allowance.

The Examiner is always welcome to telephone the attorney of record at the telephone number provided below if such telephone discussion would help expedite the prosecution of the patent application.

Prompt and favorable consideration of this Amendment is respectfully requested.

Respectfully Submitted,



Date: March 11, 2014

H C McGurk IV, Reg. No. 34,964
Attorney for Applicant

The Law Office of Clay McGurk
PO Box 1488
Orange, CA 92706
Phone: **949.395.3942**
Email: claymcgurk@gmail.com

Electronic Acknowledgement Receipt

EFS ID:	18434646
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	11-MAR-2014
Filing Date:	22-JUN-2012
Time Stamp:	15:41:01
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Response After Final Action	13531294_AAF_3-11-14.pdf	447733 5591e07fca239dd0ff8ccfeb2dd6fb24debd133e	no	10

Warnings:	
Information:	

RED HAT 1012
RED HAT V. CAS

Total Files Size (in bytes):	447733

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/531,294	Filing Date 06/22/2012	☐ To be Mailed

ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 = *		X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 = *		X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	03/11/2014	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 20	Minus	** 21	= 0	X $40 =	0
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $210 =	0
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/BRENDA J. DENNY/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RED HAT 1012
RED HAT V. CAS

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/531,294	Filing Date 06/22/2012	☐ To be Mailed

ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)		
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

		(Column 1)		(Column 2)	(Column 3)		
	03/11/2014	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
AMENDMENT	Total (37 CFR 1.16(i))	* 20	Minus	** 21	= 0	X $40 =	0
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $210 =	0
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	**0**

		(Column 1)		(Column 2)	(Column 3)		
		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
AMENDMENT	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/BRENDA J. DENNY/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

64828 7590 03/20/2014

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

EXAMINER	
CHO, HONG SOL	

ART UNIT	PAPER NUMBER
2467	

NOTIFICATION DATE	DELIVERY MODE
03/20/2014	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

CLAYMCGURK@GMAIL.COM

Advisory Action Before the Filing of an Appeal Brief	**Application No.** 13/531,294	**Applicant(s)** DELANGIS, ERIC M.	
	Examiner HONG CHO	**Art Unit** 2467	**AIA (First Inventor to File) Status** No

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 11 March 2014 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

NO NOTICE OF APPEAL FILED

1. ☒ The reply was filed after a final rejection. No Notice of Appeal has been filed. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114 if this is a utility or plant application. Note that RCEs are not permitted in design applications. The reply must be filed within one of the following time periods:
   a) ☒ The period for reply expires 3 months from the mailing date of the final rejection.
   b) ☐ The period for reply expires on: (1) the mailing date of this Advisory Action; or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.
   c) ☐ A prior Advisory Action was mailed more than 3 months after the mailing date of the final rejection in response to a first after-final reply filed within 2 months of the mailing date of the final rejection. The current period for reply expires months from the mailing date of *the prior Advisory Action* or SIX MONTHS from the mailing date of the final rejection, whichever is earlier.

   Examiner Note: If box 1 is checked, check either box (a), (b) or (c). ONLY CHECK BOX (b) WHEN THIS ADVISORY ACTION IS THE FIRST RESPONSE TO APPLICANT'S FIRST AFTER-FINAL REPLY WHICH WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. ONLY CHECK BOX (c) IN THE LIMITED SITUATION SET FORTH UNDER BOX (c). See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) or (c) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. ☐ The Notice of Appeal was filed on _____. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☒ The proposed amendments filed after a final rejection, but prior to the date of filing a brief, will not be entered because
   a) ☒ They raise new issues that would require further consideration and/or search (see NOTE below);
   b) ☐ They raise the issue of new matter (see NOTE below);
   c) ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or
   d) ☐ They present additional claims without canceling a corresponding number of finally rejected claims.

   NOTE: *See Continuation Sheet*. (See 37 CFR 1.116 and 41.33(a)).

4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).
5. ☐ Applicant's reply has overcome the following rejection(s): _____.
6. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).
7. ☐ For purposes of appeal, the proposed amendment(s): (a) ☐ will not be entered, or (b) ☐ will be entered, and an explanation of how the new or amended claims would be rejected is provided below or appended.

AFFIDAVIT OR OTHER EVIDENCE

8. ☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.
9. ☐ The affidavit or other evidence filed after final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).
10. ☐ The affidavit or other evidence filed after the date of filing the Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing of good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).
11. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

12. ☒ The request for reconsideration has been considered but does NOT place the application in condition for allowance because: Claim 17 is amended so that would require further consideration and search. It should be noted that claim 17 is not similar to claim 35. In claim 17, the first device receives the first data and the second data form one common network while in claim 35, the first device receives the first data from the first network and the second data from the second network.
13. ☐ Note the attached Information *Disclosure Statement*(s). (PTO/SB/08) Paper No(s). _____
14. ☐ Other: _____.

STATUS OF CLAIMS

15. The status of the claim(s) is (or will be) as follows:
    Claim(s) allowed: .
    Claim(s) objected to: .
    Claim(s) rejected:

/HONG CHO/
Primary Examiner, Art Unit 2467

Continuation of 3. NOTE: Amended claim 17 changes the scope of the invention.

AMENDMENT UNDER 37 C.F.R. §1.116
EXPEDITED PROCEDURE
Technology Center 2400

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Mail Stop **AF**
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

DO NOT ENTER: /H.C./ **AMENDMENT UNDER 37 C.F.R. §1.116**

In response to the final Office Action issued on December 30, 2013 ("Office Action"), please reconsider the above-identified application in light of the following amendments to the claims and the accompanying remarks.

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION

I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: March 11, 2014 By: Clay McGurk
Clay McGurk

RED HAT 1012
RED HAT V. CAS

To: CLAYMCGURK@GMAIL.COM,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

Mar 20, 2014 05:21:40 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	CTAV	03/20/2014	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL
(Submitted Only via EFS-Web)

Application Number	13531294	Filing Date	2012-06-22	Docket Number (if applicable)	ED 1040	Art Unit	2467
First Named Inventor	ERIC DELANGIS			Examiner Name	HONG CHO		

This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.
Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. The Instruction Sheet for this form is located at WWW.USPTO.GOV

SUBMISSION REQUIRED UNDER 37 CFR 1.114

Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).

☒ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.

☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____

☐ Other _____

☒ Enclosed

☒ Amendment/Reply

☐ Information Disclosure Statement (IDS)

☐ Affidavit(s)/ Declaration(s)

☐ Other _____

MISCELLANEOUS

☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of months _____ (Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)

☐ Other _____

FEES

☐ **The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.**
The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No _____

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

☒ Patent Practitioner Signature

☐ Applicant Signature

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Signature of Registered U.S. Patent Practitioner			
Signature	/H C McGurk IV, Reg No 34964/	Date (YYYY-MM-DD)	2014-03-31
Name	Harold McGurk	Registration Number	34964

This collection of information is required by 37 CFR 1.114. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Mail Stop **Amendment**
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

AMENDMENT

In response to the final Office Action issued on December 30, 2013 ("Office Action"), please reconsider the above-identified application in light of the following amendments to the claims and the accompanying remarks. A request for continued examination (RCE) is being filed concurrently with this Amendment. March 30, 2014 falls on a Sunday. Therefore the Amendment, the RCE and requisite fees are being submitted on the next business day via EFS-Web.

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION

I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: March 31, 2014 By: Clay McGurk

Clay McGurk

RED HAT 1012
RED HAT V. CAS

THE CLAIMS

This listing of claims will replace all prior versions and listings of the claims in the application:

1-16. (Canceled).

17. (Currently Amended) A method for increasing bandwidth and data throughput comprising the steps of:

establishing a wireless connection between a first device and a second device, the first device being connected at a first location to a network via a landline, the second device being connected at a second location to the network, the second location being geographically separate from the first location, the first device and the second device being configured for accessing the network; and

concurrently receiving at the first ~~location~~ device first data from the network at the first location and second data over the wireless connection from the second device, the second data being received at the second location from the network.

18. (Previously Presented) The method of claim 17, wherein the first location and the second location are physically isolated from each other.

19. (Previously Presented) The method of claim 17, further comprising the step of aggregating the second data received over the wireless connection and the first data received over the connection at the first location.

20. (Previously Presented) The method of claim 17, wherein a connection at the first location and a connection at the second location are network connection points.

21. (Previously Presented) The method of claim 17, wherein the first device and the second device are consumer networking devices.

RED HAT 1012
RED HAT V. CAS

22. (Previously Presented) The method of claim 17, further comprising the steps of; establishing a second wireless connection between the first device and a third device, the third device being connected at a third location to the network, the third location being geographically different than the first location and the second location; and

concurrently receiving third data over the second wireless connection with the receipt of the first data at the first location and the second data over the wireless connection.

23. (Previously Presented) The method of claim 22, further comprising the step of aggregating the first data, the second data and the third data.

24. (Canceled).

25. (Previously Presented) The method of claim 17, wherein the first data and the second data are received as data packets.

26. (Previously Presented) The method of claim 17, wherein the network is the Internet.

27. (Currently Amended) The method of claim 17, wherein ~~the wireless connection is established between a first wireless router at the first device and a second wireless router at~~ the second device is a wireless device.

28. (Previously Presented) The method of claim 17, wherein the network is one or a combination of a cable network, a telecommunication network, an optical network and a communications network.

29. (Previously Presented) The method of claim 17, wherein the wireless connection is a cellular connection, a satellite connection, a Wi-Fi connection, a WiMax connection or a Bluetooth connection.

30-34. (Canceled).

35. (Currently Amended) A device comprising:

an interface that establishes a landline network connection at a first location to a first network and receives first data packets over the network connection;

a wireless interface that establishes a wireless connection to a remote communications device, the remote communications device being connected at a second location to a second network, the second location being geographically separate from the first location; and

a processor that receives from the interface the first data packets from the first network and concurrently receives over the wireless interface second data packets from the second network, the second data packets being received at the second location from the second network.

36. (Previously Presented) The device of claim 35, wherein the device is one of a modem, cellular phone, a router, a computer, hotspot device and a tablet.

37. (Currently Amended) The device of claim 35, wherein the ~~the~~ device is part of a home-based network system.

38. (Canceled).

39. (Currently Amended) A method for providing broadband communications services, comprising:

receiving first data packets over a landline connection;

receiving second data packets over a wireless connection, the first data packets being concurrently received with the second data packets; and

wherein ~~aggregating~~ the first data packets create a first data stream and ~~with~~ the second data packets create a second data stream.

40. (Previously Presented) The method of claim 39, wherein the first data packets are received at a home-based network system, and wherein the second packets are received over the wireless connection at the home-based network system.

RED HAT 1012
RED HAT V. CAS

41. (Previously Presented) The method of claim 39, wherein the first data packets are received by a consumer-based network system, and wherein the second packets are received over the wireless connection at the consumer-based network system.

42. (Previously Presented) The method of claim 39, further comprising:

receiving data packets over a plurality of wireless connections, the data packets being concurrently received with the first data packets; and

aggregating the first data packets with the data packets.

43. (New) A method for increasing bandwidth and data throughput comprising the steps of:

establishing a wireless connection between a first device and a wireless device, the first device being connected at a first location to a first network, the wireless device being connected at a second location to the first network or a second network, the second location being geographically separate from the first location; and

concurrently receiving at the first device first data from the first network at the first location and second data over the wireless connection from the wireless device, the second data being received at the second location from the first network or the second network.

REMARKS

Status of the Claims

Claims 17-23, 25-29, 35-42 were pending in this application before this Amendment.

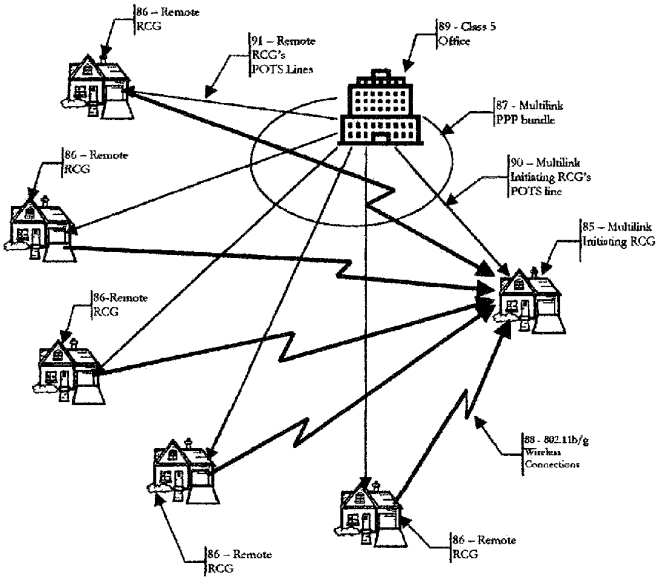
In this Amendment, claims 17, 27, 35 and 39 were amended, claim 38 was canceled and claim 43 is new.

Thus, claims 17-23, 25-29, 35-37 and 39-42 are currently pending in this application pursuant to this Amendment.

35 U.S.C. § 112 (first paragraph) Rejection

On pages 2-3 of the Office Action, claims 17-23, 25-29 and 35-42 were rejected under 35 U.S.C. §112, first paragraph. Issues of "concurrently receiving" are being raised in regards to claims 17, 22, 35, 39 and 42. "Concurrently receiving" is one the central purposes of all embodiments of the present invention - which is to receive data over many different links at the same time. This is clearly shown in Figure 7 of the present application, reproduced below:

FIG.7



The multilink initiating RCG 85 is receiving first data over POTS line 90. The five RCG's shown in Fig. 7 are each receiving their data from the network via each of their own POTS line 91. Multilink initiating RCG 85 receives first data over POTS line 90 and

RED HAT 1012
RED HAT V. CAS

concurrently receives data from remote RCGs 86 via wireless connections 88. In one embodiment of the present invention, this is referred to as a "Multilink PPP Connection" which is explained in detail in paragraphs 72-80 of the specification of the pending application, among other places. It is inherent in a "multilink PPP connection" that the multilink initiating RCG 85 receives data concurrently thereby fulfilling one of the main purposes of the invention which is to increase bandwidth for a particular location. Otherwise, if data is not being concurrently received, there is no increase in bandwidth and defeats one of the primary purposes of the present invention.

The present application incorporated by reference both the parent and grand-parent applications. The grand-parent application, serial no. 10/686,375, was published in U.S. Published Application 2005-0078690. In that original filing of the grand-parent patent application, there existed claim 3 that related to concurrent reception of data. Claim 3 of the original grand-parent application is reproduced below:

> 3. An RCG of claims 1 and 2 that dynamically allocates separate and physically diverse POTS lines or wireless connections into a multilink group capable of aggregating the combined bandwidth of a plurality of said separate physically diverse POTS lines or wireless connections and providing said aggregate bandwidth to the benefit of a single, or a plurality of RCG device for the concurrent and high speed transmission of large or multiple files; and that can utilize one or more of the separate and physically diverse POTS lines or wireless connections that are physically connected to other remote RCG devices as stand alone connections that are not grouped in a multilink configuration to the benefit of a single or a plurality of RCG devices for the concurrent high speed transmission of large or multiple files.

Fig. 7 undoubtedly shows a device that is able to connect to multiple remote devices and can concurrently receive data from each of the remote devices. Claim 3 of the original patent application confirms the "concurrent high transmission of large or multiple files." Therefore, there is clear support for what is being claimed in claims 17, 22, 35, 39 and 42. It is respectfully requested that the rejection of claims 17-23, 25-29 and 35-42 under 35 U.S.C. §112, first paragraph, be reconsidered and withdrawn.

35 U.S.C. § 112 (second paragraph) Rejections

On pages 4-5 of the Office Action, claims 40 and 41 were rejected under 35 U.S.C. §112, second paragraph as being indefinite due to issues about what is meant by a home-based network

RED HAT 1012
RED HAT V. CAS

or a consumer-based network. In paragraphs 9-10 and 12 of the present application (among others), it describes in one embodiment how the device (e.g., RCG) is simply plugged into the existing telephone line at a residence or small business. The phrases "home-based network" and "consumer-based network" should be interpreted using a plain or ordinary meaning to those of ordinary skill in the art in light of what is described in the specification. Therefore, it is respectfully requested that the rejection of claims 40 and 41 under 35 U.S.C. §112, second paragraph, be reconsidered and withdrawn.

35 U.S.C. § 103 Rejections

On pages 5-6 of the Office Action, claims 17-23, 25-29 and 35-42 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent 7,426,316 to Vehvilainen ("Vehvilainen") in view of U.S. Patent 7,113,173 to Bi *et al.* ("Bi"). Basically, Vehvilainen was relied for teaching the first paragraph of claim 17 by citing Fig. 1, col. 5, lines 11-16 and col. 4, lines 29-33. Bi was relied for teaching the second paragraph of claim 17 by citing Figs. 44 and 46, and col. 47, lines 23-35. It will be shown that the Vehvilainen and Bi in combination do not teach all the limitations of claim 17, 35 and 39.

Vehvilainen teaches four ways to transfer images or photographs which are taken on a phone 102 to a computer 104. (Vehvilainen, 5:11-16; 4:29-33). These four ways are: (1) through a communication network 103 (e.g., the Internet); or (2) a Bluetooth connection; or (3) a similar wireless connection; or (4) a cable connecting the devices. *Id.* Notably, the citations in Vehvilainen use the word "or" and never the word "and" when describing the different ways of transferring the images. By using the word "or", Vehvilainen only describes that images are transferred only over one of these four ways from the phone 102 to the computer 104, or from another phone to the phone 102. Vehvilainen never describes or suggests using two or more of these transfer methods concurrently to transfer images from the phone 102 to the computer 104. Data is sent between the phone 102 and the computer 104 over one of the connections, but not both concurrently.

Vehvilainen is using what is known to those of ordinary skill in the art as a point-to-point connection. A point-to-point connection described in Vehvilainen is the one of the problems that the present application was trying to solve. The present invention provides advantages of transferring large amounts of information or multiple files over multiple lines at the same time.

There is no description or suggestion that the phone 102 or the computer 104 of Vehvilainen is able to support concurrently sending images over different connections. However, even if Vehvilainen did suggest such, it would still fall short of what is recited in claims 17, 35 and 39. These claims require that first data is being received from the network at the first device (e.g., computer 104), ***but also*** that second data is being received by a second device (e.g., phone 102) from the network and is also transmitted wirelessly to the first device (computer 104) concurrently with the receipt of the first data. In Vehvilainen, image data ***originates*** at the phone 102, **not** from the network 103 as required in the claims. Claim 17 requires that data is received from the network 103 by the phone 102 and the computer 104. Images in Vehvilainen are transferred from the phone 102 across the network 103 to the computer 104, rather than data being transferred from the network 103 to the phone 102 then to the computer 104. Therefore Vehvilainen is missing a limitation of the claim 17 in that the phone 102 would be receiving data **from** the network and then transferring that data to computer 104.

Vehvilainen is poor prior art because this reference describes a method and apparatus for ***improving image quality***, which is non-analogous art. The methods described in Vehvilainen produce image correcting information to decrease errors in the image produced by a camera module. (Vehvilainen, Abstract). The embodiments of the claimed invention concern ways of transferring data across multiple connections at the same time, rather than ways for improving image quality such as described in Vehvilainen. Therefore, Vehvilainen is a poor reference, unrelated to what is being solved in the present application, and basically is from a different field that is non-analogous art.

Bi does not cure the deficiency of Vehvilainen because Bi is nothing more than describing a router in the portions cited in the Office Action. (Bi, 47:23-35). In Bi, Fig. 46 merely illustrates a software structure of a network system that enables a wireless device to interface with a wired LAN system such as shown in Fig. 44. That is all that Fig. 44 shows – a wireless device 101 being able to connect to a host computer system. In other words, Bi shows in Figs. 44 and 46 a way for a router to communicate with a LAN and a wireless device. The multiplexing of data from the wired LAN card 1352 and the wireless LAN card 1360 is for transmitting combined data to a network such as the Internet. *Id.* Bi simply allows a wireless device 101 to communicate with the Internet or a LAN system. Bi never describes or suggests a

way for the wireless device 101 to transmit data it receives from the network to another device as required in claims 17, 35 and 39.

A combination of Vehvilainen and Bi only results in a system where it makes possible for the phone 102 and the computer 104 to communicate with the network 103. Bi simply is describing a way that is not described in Vehvilainen for ***how*** the phone 102 connects to the network 103, or the computer 104 connects to the network 103. Therefore a combination Vehvilainen and Bi does not result in the claimed invention because both fail to describe or suggest, either individually or in combination, each and all the elements required in claims 17, 35 and 39. It is respectfully requested that the rejection of claims 17-23, 25-29 and 35-42 under 35 U.S.C. §103(a) based on Vehvilainen and Bi be reconsidered and withdrawn.

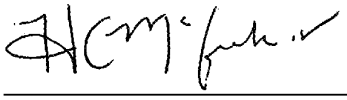
Conclusion

All rejections in the Office Action have been properly traversed. It is respectfully requested that all outstanding rejections be reconsidered and withdrawn. It is believed that a full and complete reply has been made in response to the Office Action, and, that as such, the present patent application is in a condition for allowance.

The Examiner is always welcome to telephone the attorney of record at the telephone number provided below if such telephone discussion would help expedite the prosecution of the patent application.

Prompt and favorable consideration of this Amendment is respectfully requested.

Respectfully Submitted,



Date: March 31, 2014

H C McGurk IV, Reg. No. 34,964
Attorney for Applicant

The Law Office of Clay McGurk
PO Box 1488
Orange, CA 92706
Phone: **949.395.3942**
Email: claymcgurk@gmail.com

Electronic Patent Application Fee Transmittal

Application Number:	13531294
Filing Date:	22-Jun-2012
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Filer:	Harold C. McGurk
Attorney Docket Number:	ED 1040

Filed as Small Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Independent Claims in Excess of 3	2201	1	210	210
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

RED HAT 1012
RED HAT V. CAS

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
RCE - 2nd and Subsequent Request	2820	1	850	850
		Total in USD ($)		**1060**

Electronic Acknowledgement Receipt

EFS ID:	18633603
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	31-MAR-2014
Filing Date:	22-JUN-2012
Time Stamp:	18:47:44
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$1060
RAM confirmation Number	6065
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)

RED HAT 1012
RED HAT V. CAS

1	Request for Continued Examination (RCE)	13531294_RCE_30e_fill.pdf	697841 991d93af6cb41ea5380c88efec8ccfd81e07f886	no	3
Warnings:					
Information:					
2	Amendment Submitted/Entered with Filing of CPA/RCE	13531294_RCEAmendment_3-31-2014.pdf	446930 4418fe84e8c5fdfa90e337896ef603e885d320a6	no	10
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	32045 c913bfc0f982fa6d334348503df2bb93ef753e32	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	1176816		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/531,294	Filing Date 06/22/2012	☐ To be Mailed

ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)		
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

		(Column 1)		(Column 2)	(Column 3)		
	03/31/2014	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
AMENDMENT	Total (37 CFR 1.16(i))	* 20	Minus	** 21	= 0	X $40 =	0
	Independent (37 CFR 1.16(h))	* 4	Minus	***3	= 1	X $210 =	210
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	**210**

		(Column 1)		(Column 2)	(Column 3)		
		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
AMENDMENT	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/DONNA 1. SMALLS LOGAN/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

64828 7590 04/21/2014

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

EXAMINER	
CHO, HONG SOL	

ART UNIT	PAPER NUMBER
2467	

NOTIFICATION DATE	DELIVERY MODE
04/21/2014	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

CLAYMCGURK@GMAIL.COM

Office Action Summary	Application No. 13/531,294	Applicant(s) DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03/31/2014*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5)☒ Claim(s) *17-23,25-29,35-37 and 39-43* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *17-23, 25-29, 35-37 and 39-43* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. This office action is in response to the RCE filed on 03/31/2014. Claims 17-23, 25-29, 35-37 and 39-43 are pending in the instant application.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 17, 18, 20-22, 25-29, 35-37, 39-41 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vehvilainen (US 7426316) in view of Xu (US 20090150924).

Re claims 17, 18, 20, 21, 25, 26, 35-37, 39-41 and 43, Vehvilainen discloses establishing a wireless connection between a first device (figure 1, element 104) and a second device, being a consumer device (figure 1, element 106), the first device being connected at a first location to a network (figure 1, element 103), the second device being connected at a second location to the network (figure 1; column 5, lines 11-16), the

RED HAT 1012
RED HAT V. CAS

second location being geographically separate from the first location, the first device and the second device being configured for accessing the network (column 4, lines 29-33). Vehvilainen discloses the second device can receive images produced by the camera module over the Internet (column 4, lines 29-33), but fails to disclose concurrently receiving data from the Internet and the wireless connection. However, Xu discloses simultaneously receiving packets from different transmission medium (paragraph [0023]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Vehvilainen with the teaching of Xu so that the first device receives first data from the Internet and second data from the second device, where the second device receives data produced by the camera module over the Internet.

Re claim 18, Vehvilainen discloses the first location and the second location are physically isolated from each other (figure 1).

Re claim 22, Vehvilainen discloses establishing a second wireless connection between the first device and a third device (column 4, lines 29-33), but fails to disclose concurrently receiving third data over the second wireless connection with the receipt of the first data at the first location and the second data over the wireless connection. However, Xu discloses simultaneously receiving packets from different transmission media (paragraph [0023]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Vehvilainen with the teaching of Xu for the benefit of efficient utilization of network resources by multiplexing data received from different transmission media.

RED HAT 1012
RED HAT V. CAS

Re claims 27-29, Vehvilainen discloses the wireless connection is established between a first wireless router at the first device and a second wireless router at the second device, wherein the network is one or a combination of a cable network, a telecommunication network, an optical network and a communications network and the wireless connection is a cellular, satellite, Wi-Fi, WiMax or Bluetooth connection (column 5, lines 11-16).

4. Claims 19, 23 and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vehvilainen in view of Xu and Jost et al (US 7653089, "Jost").

Re claims 19, 23 and 42, Vehvilainen discloses all of the limitations of the base claim, but fails to disclose aggregating the first data, the second data and the third data.

However, it is well known in the art that traffic flows received from a plurality of senders are multiplexed. Jost discloses multiplexing data streams from various service providers (figure 7). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Vehvilainen with the teaching of Jost for the benefit of providing efficient utilization of network resources.

Response to Arguments

5. Applicant's arguments with respect to claims have been considered but are moot in view of the new ground(s) of rejection.

RED HAT 1012
RED HAT V. CAS

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hong Cho whose telephone number is 571-272-3087. The examiner can normally be reached on Mon-Fri during 7 am to 4 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hassan Phillips can be reached on 571-272-3940. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Hong Cho/

Primary Examiner, Art Unit 2467

RED HAT 1012
RED HAT V. CAS

Notice of References Cited	Application/Control No. 13/531,294	Applicant(s)/Patent Under Reexamination DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-2009/0150924	06-2009	Xu, Yue Heng	725/31
*	B	US-7,653,089	01-2010	Jost et al.	370/485
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	11/26/2012	05/21/2013	09/05/2013	12/23/2013	04/16/2014				
	1	✓	✓	-	-	-				
	2	✓	✓	-	-	-				
	3	✓	✓	-	-	-				
	4	✓	✓	-	-	-				
	5	✓	✓	-	-	-				
	6	✓	✓	-	-	-				
	7	✓	✓	-	-	-				
	8	✓	✓	-	-	-				
	9		✓	-	-	-				
	10		✓	-	-	-				
	11		✓	-	-	-				
	12		✓	-	-	-				
	13		✓	-	-	-				
	14		✓	-	-	-				
	15		✓	-	-	-				
	16		✓	-	-	-				
	17			✓	✓	✓				
	18			✓	✓	✓				
	19			✓	✓	✓				
	20			✓	✓	✓				
	21			✓	✓	✓				
	22			✓	✓	✓				
	23			✓	✓	✓				
	24			✓	-	-				
	25			✓	✓	✓				
	26			✓	✓	✓				
	27			✓	✓	✓				
	28			✓	✓	✓				
	29			✓	✓	✓				
	30			✓	-	-				
	31			✓	-	-				
	32			✓	-	-				
	33			✓	-	-				
	34			✓	-	-				
	35			✓	✓	✓				
	36			✓	✓	✓				

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	11/26/2012	05/21/2013	09/05/2013	12/23/2013	04/16/2014				
	37			✓	✓	✓				
	38				✓	-				
	39				✓	✓				
	40				✓	✓				
	41				✓	✓				
	42				✓	✓				
	43					✓				

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S174	26	S173 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2014/04/16 07:56
S173	96	(multiplex$3 or aggregat$3 or combin$3) with (data or packet$1 or frame$1) near wired with wireless$2	US-PGPUB; USPAT	OR	OFF	2014/04/16 07:56
S176	10	(eric near delangis).in.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2014/04/16 08:17
S175	69696	(370/235,230,233,363,371,374,381,433,465-468,470-474,483-486,493-495,352,395,252).ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2014/04/16 08:17
S177	3	concurrently near receiv$3 near (data or packet$1 or frame$1) same ((wired or landline) with wireless)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2014/04/16 08:19
S178	36	(multiplex$3 or aggregat$3 or combin$3) with received near (data or packet$1 or frame$1) with wired with wireless$2	US-PGPUB; USPAT	OR	OFF	2014/04/16 08:21
L6	499	receiv$3 near data with wirelessly with server	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2014/04/16 09:53
L7	46	l6 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2014/04/16 09:54
L22	3	multi near link with concurrent$2 with (receiv$3 or send$3 or transmit$4)	US-PGPUB; USPAT	OR	OFF	2014/04/16 10:36
L45	58	l44 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2014/04/16 11:25
L44	309	(concurrent$2 or simultaneous$2) with receiv$3 with transmission near (media or medium)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2014/04/16 11:25

L46	92	(multiplex$3 or aggregat$3 or combin$3) with (data or packet$1 or frame$1) with received with wired with wireless$2	US-PGPUB; USPAT	OR	OFF	2014/04/16 12:18
L47	15	l46 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2014/04/16 12:19
L54	4	09/953980	US-PGPUB; USPAT	OR	OFF	2014/04/16 13:17
L63	492	(multiplex$3 or aggregat$3 or combin$3) with received near (traffic or data or packet$1 or frame$1) near (stream or flow)	US-PGPUB; USPAT	OR	OFF	2014/04/16 14:06

4/16/2014 2:22:06 PM
C:\Users\hcho\Documents\EAST\Workspaces\13531294.wsp

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	Art Unit
	HONG CHO	2467

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
370	230-235,278,282,328,338,468,477	11/26/2012	hc

SEARCH NOTES		
Search Notes	Date	Examiner
EAST text search	11/26/2012	hc
updated search	5/21/2013	hc
updated search	9/5/2013	hc
updated search	12/23/2013	hc
updated search	4/16/2014	hc

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

To: CLAYMCGURK@GMAIL.COM,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

Apr 21, 2014 05:18:40 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	CTNF	04/21/2014	ED 1040
	892	04/21/2014	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE
COMMISSIONER FOR PATENTS
P.O.BOX 1450
ALEXANDRIA VA 22313-1451

PRESORTED
FIRST-CLASS MAIL
U.S. POSTAGE PAID
POSTEDIGITAL
NNNNN

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

Courtesy Reminder for
Application Serial No: 13/531,294

Attorney Docket No: ED 1040
Customer Number: 64828
Date of Electronic Notification: 04/21/2014

This is a courtesy reminder that new correspondence is available for this application. If you have not done so already, please review the correspondence. The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

An email notification regarding the correspondence was sent to the following email address(es) associated with your customer number:
CLAYMCGURK@GMAIL.COM

To view your correspondence online or update your email addresses, please visit us anytime at **https://sportal.uspto.gov/secure/myportal/privatepair.** If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov or call 1-866-217-9197.

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	13531294
Filing Date	2012-06-22
First Named Inventor	Eric Delangis
Art Unit	2467
Examiner Name	Hong Cho
Attorney Docket Number	ED 1040

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] ¡	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	13531294
	Filing Date	2012-06-22
	First Named Inventor	Eric Delangis
	Art Unit	2467
	Examiner Name	Hong Cho
	Attorney Docket Number	ED 1040

	1	H. Balakrishnan et al., "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", IEEE/ ACM Transactions on Networking, 17 pages; 1997.	☐
	2	J. Chen et al., "An Efficient Multipath Forwarding Method", Proceedings of IEEE Infocom '98; 9 pages; 1998.	☐
	3	J. Chen, "New Approaches to Routing for Large-Scale Data Networks", Thesis for Doctor of Philosophy, Rice University; 259 pages; 1999.	☐
	4	M. Handley et al., "Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics", Proceedings of the 10th USENIX Security Symposium; 18 pages; 2001.	☐
	5	H. Hsieh et al., "A Transport Layer Approach for Achieving Aggregate Bandwidths on Multi-homed Mobile Hosts", Proc. of MobiCom; 12 pages; August 2002.	☐
	6	E. Kohler et al., "The Click Modular Router", Proc. of the 17th Symposium on Operating Systems Principles; 34 pages; 1999.	☐
	7	S. Lee et al., "Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks", Infocom 98 - Proceedings, IEEE (Volume 3); 5 pages; 1998.	☐
	8	Y. Lee et al., "Improving TCP Performance in Multipath Packet Forwarding Networks", Journal of Communications and Networks, Vol. 4, No. 2; 10 pages; June 2002.	☐
	9	L. Magalhaes et al., "Transport Level Mechanisms for Bandwidth Aggregation on Mobile Hosts", IEEE-ICNP; 7 pages; 2001.	☐
	10	S. Mascolo et al., "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", Proc. of the ACM Mobicom; 11 pages; 2001.	☐
	11	J. Mo et al., "Analysis and Comparison of TCP Reno and Vegas", University of California at Berkeley; 19 pages; 1998.	☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	13531294
	Filing Date	2012-06-22
	First Named Inventor	Eric Delangis
	Art Unit	2467
	Examiner Name	Hong Cho
	Attorney Docket Number	ED 1040

	12	J. Mo et al., "Analysis and Comparison of TCP Reno and Vegas", IEEE; 8 pages; 1999.	☐
	13	P. Rodriguez et al., "Dynamic Parallel Access to Replicated Content in the Internet", IEEE/ACM Transactions on Networking; 11 pages; 2000.	☐
	14	P. Srisuresh et al., "Request for Comments: 2663 - IP Network Address Translator (NAT) Terminology and Considerations"; The Internet Society; 30 pages; 1999.	☐
	15	E. Steinbach et al., "A Simulation Study of Packet Path Diversity for TCP File Transfer and Media Transport on the Internet", Stanford University; 8 pages; August 2002.	☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	13531294
	Filing Date	2012-06-22
	First Named Inventor	Eric Delangis
	Art Unit	2467
	Examiner Name	Hong Cho
	Attorney Docket Number	ED 1040

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☒ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☐ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/H C McGurk IV/	Date (YYYY-MM-DD)	2014-06-03
Name/Print	H C McGurk	Registration Number	34964

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT
TRANSMITTAL LETTER

This submission does not represent that a search has been made or that no better art exists and does not constitute an admission that each or all of the listed documents are material or constitute prior art. If the Examiner applies any of the documents as prior art against any of the claims in the present application, all rights are reserved to take appropriate action to establish the patentability of the disclosed invention over the present application.

Best Regards,

Date: June 3, 2014

H C McGurk IV, Reg. No. 34,964
Attorney for Applicant

The Law Office of Clay McGurk
PO Box 1488
Orange, CA 92856
Phone: **949.395.3942**

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION

I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: March 30, 2012

By: Clay McGurk

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

RED HAT 1012
RED HAT V. CAS

RICE UNIVERSITY

New Approaches to Routing for Large-Scale Data Networks

by

Johnny Chen

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Dr. Peter Druschel, Chair
Assistant Professor in Computer Science

Dr. Devika Subramanian
Associate Professor in Computer Science

Dr. Edward Knightly
Assistant Professor in Electrical Engineering

Houston, Texas

June, 1999

RED HAT 1012
RED HAT V. CAS

UMI Number: 9969239

UMI®

UMI Microform 9969239

Copyright 2000 by Bell & Howell Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

RED HAT 1012
RED HAT V. CAS

New Approaches to Routing for Large-Scale Data Networks

Johnny Chen

Abstract

This thesis develops new routing methods for large-scale, packet-switched data networks such as the Internet. The methods developed increase network performance by considering routing approaches that take advantage of more available network resources than do current methods. Two approaches are explored: dynamic metric and multipath routing. Dynamic metric routing provides paths that change dynamically in response to network traffic and congestion, thereby increasing network performance because data travel less congested paths. The second approach, multipath routing, provides multiple paths between nodes and allows nodes to use these paths to best increase their network performance. Nodes in this environment achieve increased performance through aggregating the resources of multiple paths.

This thesis implements and analyzes algorithms for these two routing approaches. The first approach develops hybrid-Scout, a dynamic metric routing algorithm that calculates *independent* and *selective* dynamic metric paths. These two calculation properties are key to reducing routing costs and avoiding routing instabilities, two difficulties commonly experienced in traditional dynamic metric routing. For the second approach, multipath routing, this thesis develops a complete multipath network that includes the following components: routing algorithms that compute multiple paths, a multipath forwarding method to ensure that data travel their specified paths, and an end-host protocol that effectively uses multiple paths.

Simulations of these two routing approaches and their components demonstrate signif-

RED HAT 1012
RED HAT V. CAS

icant improvement over traditional routing strategies. The hybrid-Scout algorithm requires 3-4 times to 1-2 orders of magnitude less routing cost compared to traditional dynamic metric routing algorithms while delivering comparable network performance. For multi-path routing, nodes using the multipath protocol fully exploit the offered paths and increase performance linearly in the additional resources provided by the multipath network. The performance improvements validate the multipath routing algorithms and the effectiveness of the proposed end-host protocol. Furthermore, this new multipath forwarding method allows multipath networks to be supported at low routing costs. This thesis demonstrates that the proposed methods to implement dynamic metric and multipath routing are efficient and deliver significant performance improvements.

RED HAT 1012
RED HAT V. CAS

Acknowledgments

This thesis is the culmination of many years of academic and personal education. I would like to take this opportunity to thank the many people who have made significant contributions to my educational process.

First and foremost, I am indebted to my parents. I am especially grateful to my mother, who made countless sacrifices to ensure that her children receive the best opportunity to live full and joyful lives. My brother Steve and sisters Aileen and Jennifer have also given me an immeasurable amount of fun, love, and advice. My family's constant support provided a level of personal and educational development that I would otherwise never have received. This thesis is a tribute to the positive influences they have made in my life.

I also thank my advisors Peter Druschel and Devika Subramanian. Their high standards of research sharpened not only my problem solving skills, but also my ability to effectively convey technical and complex ideas. In addition, they gave me complete freedom to pursue my research interests and were open to projects outside their core research areas. Of course, this thesis would not have been possible without my advisors' help and guidance. I also thank my third committee member, Ed Knightly, for his assistance in my research.

My high school teachers Mrs. Cynthia Erdei, Mr. Lynn Rosier, and Mrs. Ann Koenig played a significant role in my decision to pursue higher education. Their dedication to teaching and challenging young minds opened doors for me to attend college. I wish for their continuing success in teaching and positively shaping young lives.

I am grateful for the friends who have enriched my life: friends Bo and Jerry, and college friends Judd, Cash, Jeff, Eddie, and Deanna. These friendships have become stronger over the years and will hopefully continue to develop. I have also made many life-long friends while at Rice, including lunch buddies Zoran and Dejan; officemates Zhenghua,

RED HAT 1012
RED HAT V. CAS

Mohit, and Henrik; and good friends Cristiana and Phil. Others friends include Anita, Gaurav, Honghui, Weimin, Bo, Karthick, Ram, Mike, Collin, James, Shriram, Robby, John, Matthew, Cathy, Fay, Darren, and Mary. Each one of these friends has made my life in Houston very enjoyable.

Last but not least, I thank my writing instructor Dr. Jan Hewitt for significantly improving my writing skills and her tireless evaluations of my thesis.

RED HAT 1012
RED HAT V. CAS

Contents

Abstract ii

Acknowledgments iv

List of Illustrations xi

List of Tables xiv

1 Introduction 1

1.1 Contributions 6

1.2 Dissertation Overview 9

2 Background 10

2.1 The Distance Vector Routing Algorithm 10

2.2 The Link State Routing Algorithm 13

2.3 The Internet 14

2.4 Comparison of Routing Models 17

2.4.1 Static and Dynamic Metrics 17

2.4.2 Resource Usage 20

2.4.3 Granularity 24

2.4.4 Routing Model Summary 25

3 Dynamic Metric Routing 27

3.1 The Hybrid Approach 30

3.1.1 The Scout Algorithm 31

3.1.2 Dynamic Metric Scout 34

3.1.3 Scout-DV Hybrid 36

RED HAT 1012
RED HAT V. CAS

3.1.4 Scout-LS Hybrid 42
3.1.5 Algorithm Comparisons 46
3.2 Simulation Results 47
3.2.1 Simulation Environment 48
3.2.2 A Forest Topology 50
3.2.3 Background Traffic 57
3.2.4 Foreground Traffic 60
3.2.5 Path Approximations 62
3.3 Hybrid-Algorithm Summary 63

4 Multipath Routing **65**
4.1 Multipath Routing Definitions 65
4.2 Multipath Routing Overview 69
4.2.1 Multipath Advantages 70
4.2.2 Multipath Disadvantages 72
4.2.3 Multipath Implementation Cost 75
4.2.4 Multipath Benefits 77
4.2.5 Static and Dynamic Metric Multipath Routing 81
4.3 Multipath Routing Summary 82

5 Path Calculation Algorithms **84**
5.1 Path Characteristics 84
5.2 Path Calculation Algorithms 87
5.2.1 Minimizing Delay 88
5.2.2 Maximizing Throughput 91
5.3 Path Calculation Summary 94

6 Multipath Forwarding **96**
6.1 The Multipath Forwarding Problem 97

RED HAT 1012
RED HAT V. CAS

6.1.1 Multi-Service Path Forwarding 99
6.1.2 Multi-Option Path Forwarding 101
6.1.3 Suffix Matched Path Sets 102
6.2 Distance Vector Extension 105
6.2.1 Methods of Calculating Multiple Paths in DV 106
6.2.2 Multipath DV Extensions 107
6.2.3 MPDV Example 108
6.3 Link State Extension 110
6.3.1 Multipath LS Extensions 111
6.3.2 Example of LS Extension 112
6.3.3 Suffix Matched Multipath Routing Algorithms 112
6.3.4 Non-Suffix Matching Paths in LS 113
6.4 A Multipath Forwarding Example 116
6.5 Multipath Forwarding Summary 117

7 Multipath Transport Protocol 119

7.1 Usage Layer 120
7.2 Throughput Optimization 122
7.3 TCP 124
7.4 MPTCP 126
7.4.1 The MPTCP Algorithm 126
7.4.2 MPTCP Fairness 128
7.4.3 Path Information 130
7.4.4 Limitations 131
7.5 MPTCP Experiments 132
7.5.1 Aggregate Throughput 133
7.5.2 MPTCP Congestion Backoff Percentage 136
7.5.3 Foreground and Background Traffic 138

RED HAT 1012
RED HAT V. CAS

7.5.4 Network Resources and Throughput 142
7.6 MPTCP Summary 143

8 Multipath Routing Algorithms 145
8.1 The MPDV Capacity Removal Algorithm 145
8.1.1 Data Structure Costs 146
8.1.2 The MPDV Algorithm 147
8.1.3 MPDV Capacity Removal Example 150
8.1.4 Capacity Removal MPDV Costs 152
8.2 The MPLS Capacity Removal Algorithm 153
8.2.1 The MPLS Algorithm 153
8.2.2 Basic MPLS Forwarding 154
8.2.3 Non-Suffix Matched Paths 155
8.2.4 Capacity Removal MPLS Costs 157

9 Multipath Cost-Benefit Analysis 160
9.1 Simulation Environment 161
9.2 Throughput Performance 162
9.2.1 Basic Throughput Performance 163
9.2.2 MPLS and MPDV Throughput Differences 165
9.2.3 MPLS and MPDV Path Calculation Process 167
9.2.4 MPLS and MPDV Throughput Summary 168
9.3 Latency and Message Drop Performance 169
9.3.1 The Multipath Ping Program 170
9.3.2 Round-trip Latency 171
9.3.3 Message Drop Probability 173
9.3.4 Latency and Drop Probability Summary 174
9.4 Routing Costs 175
9.4.1 Per Packet Forwarding Overhead 176

RED HAT 1012
RED HAT V. CAS

9.4.2 Router Storage Overhead 179
9.4.3 CPU Usage in Path Computation 184
9.4.4 Routing Message Cost 185
9.4.5 Routing Cost Summary 188
9.5 Experimental Conclusions 191

10 Related Work **193**
10.1 Dynamic Metric Routing 193
10.2 Multipath Routing 198
10.2.1 Multipath Networks and Architectures 198
10.2.2 Multipath Calculation Algorithms 202
10.2.3 Multipath Forwarding Methods 204

11 Conclusions and Future Directions **207**
11.1 Conclusions 207
11.2 Future Directions 210
11.2.1 Hybrid-Scout 210
11.2.2 Multipath Routing 211

A The Scout Routing Algorithm **214**
A.1 Flood Based Route Discovery 214
A.2 Scout Proof of Correctness 220
A.3 Scout Summary 222

Bibliography **224**

RED HAT 1012
RED HAT V. CAS

Illustrations

1.1 The static metric single shortest path routing model. 2
1.2 A conceptual diagram of the dynamic metric routing model. 4
1.3 A conceptual diagram of the static metric multiple path routing model. . . . 5

3.1 Traffic locality in dec-pkt-4 TCP packet traces from *http://ita.ee.lbl.gov/html/contrib/*. 29
3.2 An example of the Scout routing algorithm. 33
3.3 Another example of the Scout routing algorithm. 34
3.4 An example of a path that the DV component calculates to a Scout generating destination after the failure of link (R, P). 41
3.5 The forest topology used in the hybrid Scout experiments. 51
3.6 The performance graphs for dynamic metric DV, LS and hybrid Scout algorithms. 52
3.7 The cost graphs for dynamic metric DV, LS and hybrid Scout algorithms. . 52
3.8 Comparative performance of hybrid Scout, LS and DV for 1 and 5 hot destinations. 53
3.9 The foreground and background performance in a large network. 58
3.10 Routing costs of different dynamic metric routing algorithms. 59
3.11 Network performance and routing cost. 60
3.12 Routing performance and cost using hold-downs. 63

4.1 An example network topology. 71
4.2 A simple three node network with full-duplex links. 78

RED HAT 1012
RED HAT V. CAS

4.3 Simulation results obtained in a simple three node network. 79
4.4 Performance of Single path TCP versus MPTCP. 80

5.1 An Example Network Topology. 86
5.2 The pseudocode for the discount shortest path algorithm. 92
5.3 The pseudocode for the capacity removal algorithm. 95

6.1 A forwarding example in a multi-service single-option multipath network. . 100
6.2 A forwarding example in a single-service multi-option multipath network. . 102
6.3 Example of MPDV. 109
6.4 Example of the multipath Link State Algorithm. 112
6.5 Example forwarding tables in a network with multi-service and multi-option paths. 116

7.1 A simple three node network. 122
7.2 MPTCP connection using three paths. 128
7.3 Non-congestion aware multipath protocol versus MPTCP on a triangle network. 134
7.4 Congestion backoff percentage on a three node network. 137
7.5 The foreground and background performance of MPTCP using the capacity removal algorithm. 139
7.6 MPTCP throughput on a 20 node network with varying network connectivity. 143

8.1 The pseudocode for the MPDV capacity removal algorithm. 149
8.2 An example of the capacity removal MPDV algorithm. 151
8.3 An example of the capacity removal MPLS non-suffix matched forwarding. 156

RED HAT 1012
RED HAT V. CAS

9.1 The foreground and background MPTCP performance using the MPLS capacity removal algorithm. 164
9.2 The foreground and background MPTCP performance using MPDV capacity removal algorithm. 164
9.3 The foreground performance of MPTCP and SPTCP using the capacity removal MPLS and MPDV algorithms on a sparse network topology. 166
9.4 The number of path calculated by each the MPDV and MPLS algorithms in a sparse network topology. 167
9.5 An example of the capacity removal path calculation to $N1$. 168
9.6 The percentage of non-suffix matched paths calculated by MPLS in a cluster topology. 169
9.7 The measured round-trip latency observed by the MP_ping and SP_ping programs. 172
9.8 The measured ping drop percentages of MP_ping and SP_ping. 173
9.9 An example of a non-suffix matched path set. 180
9.10 The aggregate forwarding table storage for MPLS and MPDV in a cluster network. 181
9.11 The total general storage cost for MPLS and MPDV in a cluster network. . 183
9.12 The routing message cost for capacity removal MPLS and MPDV algorithms. 186
9.13 The routing cost of MPDV and MPLS for a single link failure and recovery. 187

A.1 Example network topology and the first round broadcast tree from N1. . . . 216
A.2 Broadcast tree after the second round of flooding. 218
A.3 Scout Algorithm Summary for Unrestricted Networks 219

RED HAT 1012
RED HAT V. CAS

Tables

3.1 The scalability characteristics of hybrid-Scout, Distance Vector, and Link State routing algorithms. 55

9.1 A summary of the routing costs incurred by different routing algorithms. . . 191

RED HAT 1012
RED HAT V. CAS

Chapter 1

Introduction

Large-scale, wide area data networks are a part of today's global communication infrastructure. Networks such as the Internet have become an integral medium of information transfer, ranging from personal communication to electronic commerce and entertainment. The importance of such networks will only increase as the electronic world becomes more prevalent.

The basic function of a data network is very simple: delivering data from one network node to another. Achieving this goal requires many network components, including physical computers and links, signaling protocols between computers, and data packaging protocols. This thesis addresses one such component, routing, the process that logically connects network nodes by calculating paths between nodes so that data sent by one node traverses the calculated path to its destination.

Although many algorithms in graph and operational research literature calculate paths between nodes, the challenge in developing network routing algorithms is in dealing with the scale and distribution of the physical network. Because typical wide area networks have nodes on the order of tens of thousands, routing algorithms must be scalable. In addition, routing algorithms must be able to calculate paths in a distributed manner due to the global and distributive nature of physical networks. Moreover, because of the actual physical network, routing algorithms need to cope with events such as physical component failures and recalculate paths whenever such events occur. Finally, routing algorithms need to calculate paths to allow nodes to achieve high network performance.

In general, routing algorithms view a network as a weighted graph, where network links are represented as graph edges and network routers as graph vertices. Network routers are

RED HAT 1012
RED HAT V. CAS

network nodes that execute routing algorithms and ensure that data travel the calculated paths. In the weighted graph, the assignment of edge weights depends on the specific routing algorithm; typically, the assignment reflects the latency and bandwidth of the link [94]. After a routing algorithm makes these link cost assignments, it then computes paths between nodes. Thus, the specific routing algorithm that routers execute determines the paths that data will travel in the network.

Routing algorithms in today's Internet base their implementations on the *static metric single shortest path* routing model. Single shortest path means that routing algorithms provide, at any given time, the least-cost path between nodes. Static metric refers to link cost assignments which are based on static properties of a link, such as its bandwidth or latency. As shown later, the main drawback of this model is that static metric shortest paths do not always provide good network performance.

Although Internet routing algorithms use static metrics, this does not imply that the paths themselves are static. On the contrary, current Internet routing algorithms are *adaptive*, meaning they are able to recompute paths and reroute packets when network components (nodes or links) fail or recover. Therefore, even if routers or links fail, as long as a path exists between a node pair, Internet routing algorithms ensure that these two nodes can communicate with each other. Figure 1.1 shows the conceptual Internet routing model.

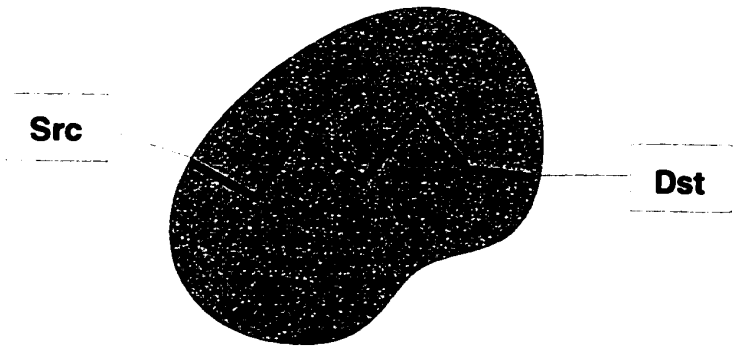


Figure 1.1 : The static metric single shortest path routing model. The solid line denotes the single path provided between Src and Dst.

RED HAT 1012
RED HAT V. CAS

The solid line in Figure 1.1 shows the shortest path calculated between Src and Dst. The shaded area denotes the network resources (network links and routers) that are not allocated for this communication channel. In this figure, messages sent by Src to Dst travel the solid path. Since the shortest path is calculated between nodes, packets travel to their destinations using paths that minimize the statically assigned cost. If link costs are uniform, then the shortest path minimizes the number of links and routers traversed. In general, a link's cost describes some notion of performance, usually given as a combination of the link's delay and bandwidth. For example, in the Internet, a link has a lower cost compared to the cost of another link with higher delay and lower bandwidth. Here, a link's cost reflects a notion of performance in terms of the amount of time expected for packets to traverse the link. With this cost assignment (expressed in terms of path delay and bandwidth), the shortest path minimizes the "expected time" for packets to reach their destinations.

However, performance measures based on the time a packet reaches its destination depend not only on the static properties of the links the packet traverses, but also on those links' utilization levels. For example, it takes less time for a packet to traverse a path when the path's links are idle than when the links are *congested*[*]. In fact, work by Khanna and Zinky [94] showed that computing paths that factor the dynamics of link utilization has a large impact on network performance.

To provide better network performance, methods developed in this thesis implement two routing models that deviate from the static metric single shortest path routing model. These two models are *dynamic metric single path routing* and *static metric multipath routing*. The methods developed improve upon previous methods that implement the two routing models. Conceptually, the two routing models increase network performance by effectively utilizing currently unallocated network resources (links and routers). These un-

[*]A link is in a congested state when too many messages are contending for the link's resources (e.g. link bandwidth and allocated router buffer space for the link). In this scenario, the router buffering this link's out-going data may be forced to drop messages out-going on the link.

RED HAT 1012
RED HAT V. CAS

allocated resources are represented by the shaded region in Figure 1.1.

The first model, dynamic metric single path routing, provides one path between node pairs, where the path computed considers network traffic and congestion. Dynamic metrics are defined as link cost metrics that change dynamically. In the context of this thesis, a link's dynamic cost is based on the link's utilization. With this type of metric, a link's cost is higher when it is experiencing congestion than when it is idle. In the dynamic metric routing model, routers recompute least-cost paths between nodes in response to dynamic changes in link costs. Because dynamically least-cost paths consider network traffic, dynamic metric routing offers the ability to provide nodes with higher performance paths (in terms of lower delay and/or higher throughput). Figure 1.2 shows a conceptual diagram of the dynamic metric single path routing model.

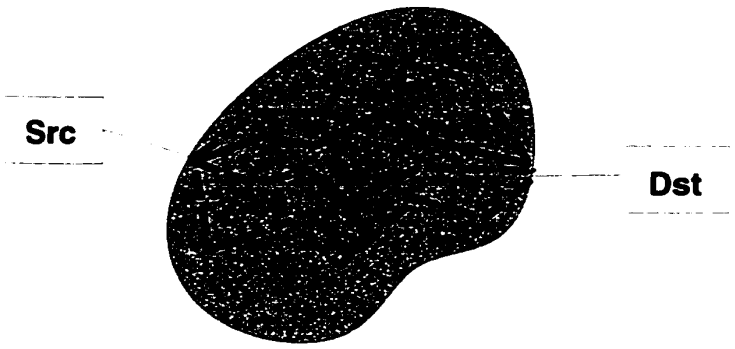


Figure 1.2 : A conceptual diagram of the dynamic metric routing model. The solid line denotes the current least-cost path between Src and Dst, and the dotted lines represent potential least-cost paths depending on dynamic link costs.

In this figure, the solid line represents the current path a dynamic metric routing algorithm provides between Src and Dst, and the three dotted lines denote potential paths Src has to Dst, depending on different traffic conditions. In a dynamic metric single path routing model, Src has available, at any one time, only one path to Dst. This path is chosen by the routing algorithm to carry data between Src and Dst. Because the routing algorithm continually recomputes the least-cost dynamic metric paths, the chosen path between nodes

RED HAT 1012
RED HAT V. CAS

reflects the current least-cost path. Notice that the path Src has to Dst depends on network traffic and the routing algorithm's recomputation speed.

The second routing model considered in this thesis, the static metric multipath routing model, deviates from static metric single shortest path by providing multiple paths between nodes. Routing algorithms in this model provide potentially multiple paths between nodes concurrently, thereby increasing a node's available resources and allowing the node to use the multiple paths in ways that best increase its performance. Figure 1.3 shows the conceptual diagram of this routing model.

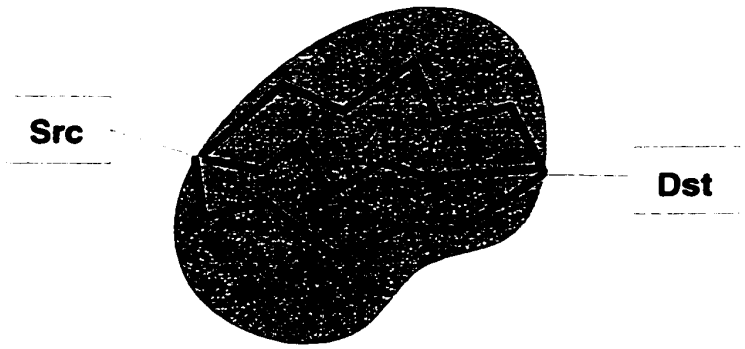


Figure 1.3 : A conceptual diagram of the static metric multiple path routing model. The three solid lines denote the paths that a multipath network provides between Src and Dst. Here Src can send data to Dst using any of the three paths.

In Figure 1.3, the solid lines represent the available paths Src has to Dst. In this example, a multipath routing algorithm provides three paths between Src and Dst. Additionally, the model allows Src to decide how to send data on these paths. For example, Src can send data to Dst on all three paths simultaneously, one path at a time, or any combination it chooses. This model allows higher performance because Src can dynamically detect and use path(s) that best maximize Src's performance.

The ability to choose which path to use differentiates the solid lines in multipath routing and the dotted lines in dynamic metric routing (for convenience, we use "dynamic metric routing" instead of "dynamic metric single path routing" and "multipath routing" instead

RED HAT 1012
RED HAT V. CAS

of "static metric multipath routing", unless otherwise specified). In multipath routing, the decision of which path to use is delegated to the sending nodes (end-hosts or the applications running on the hosts). That is, multipath routing algorithms calculate multiple paths between nodes, and the end-hosts choose which path(s) to use. In contrast, dynamic metric routing algorithms provide, at any give time, the only one path between node pairs. Thus, end-hosts in this environment do not have the option to choose among the paths the routing algorithm might calculate.

Although the general multipath routing model can also use dynamic metrics (the dynamic metric multipath model provides multiple paths between nodes that consider traffic patterns), this thesis primarily focuses on the static metric multipath routing model. The reason is that if a static metric multipath routing algorithm calculates the appropriate paths *and* if end-hosts manage these paths effectively, then the situations where dynamic metrics can benefit are significantly reduced. For example, assume that in Figure 1.2, a multipath routing algorithm provides all four paths (both dotted and solid paths) between Src and Dst. In this case, even if the multipath routing algorithm uses dynamic metrics, the opportunity to recalculate paths are much reduced because the possible dynamic metric candidate paths are already provided. Moreover, if end-hosts effectively use multiple paths to avoid congestion, then link congestion is less likely to occur which, in turn, reduces path recomputations. This issue is discussed in detail in Section 4.2.5.

Notice that the two routing models described are still adaptive to network changes. That is, both dynamic metric and multipath routing models recalculate paths upon detection of network component failures or recoveries. Therefore, methods proposed in this thesis are adaptive in that they offer the same connectivity guarantees as the current Internet routing model, but they promise higher achievable network performance.

1.1 Contributions

The contributions of this thesis stem from the development of methods that implement the dynamic metric and multipath routing models. For dynamic metric routing, a novel algo-

RED HAT 1012
RED HAT V. CAS

rithm is developed which outperforms previous dynamic metric algorithms and requires lower routing costs. For multipath routing, the major contributions are an efficient path forwarding method and a multipath transport protocol that maximizes end-to-end throughput. The contributions of this thesis are summarized below.

Contribution 1: A novel dynamic metric routing algorithm that outperforms previous dynamic metric routing methods.

Dynamic metric routing has been implemented in the past; however, previous implementations suffer two problems: routing instability and high routing overheads [94]. Routing instability occurs when paths are constantly being recomputed and do not stabilize. This instability degrades network performance and increases the probability of network congestion and failures. The second problem is high routing overheads. The overheads are in terms of the CPU cycles and messages required for path recomputation. Because routing overheads in traditional dynamic metric algorithms depend on network traffic patterns, mechanisms such as thresholding and hold-downs [74] are needed to limit the amount of routing overhead; however, the effects on route quality of these two mechanisms are uncertain because routers often compute paths based on out-dated information of the network state.

The new dynamic metric routing algorithm developed in this thesis reduces the above problems. The algorithm, hybrid-Scout, reduces routing instabilities by using two techniques: time staggered and selective dynamic metric path computation. Simulations show that hybrid-Scout reduces the amount of routing cost by as much as an order of magnitude while maintaining performance comparable to traditional dynamic metric routing algorithms. This new algorithm promises to provide the benefits of dynamic metric routing without the undesirable side-effects.

Contribution 2: An efficient multipath forwarding method.

One of the main challenges of multipath routing is the forwarding of data on their intended paths. Multipath forwarding is difficult because each node has potentially multiple paths to a destination; therefore nodes have to label data packets to indicate which path a

RED HAT 1012
RED HAT V. CAS

packet should travel. The *path forwarding problem* is defined as how to specify a packet's path and forward the packet along the specified path.

This thesis presents a novel multipath forwarding method to solve the path forwarding problem by using small, fixed-length path identifiers (IDs). This method guarantees path forwarding by chaining path ID agreements along paths [3, 85]. To do this with minimal computation and message overhead, the method requires that the calculated path set satisfy a specific property, called the *suffix matched* property. For path sets that do not satisfy this property, efficient closure operators are developed to convert them into suffix matched sets.

The novel forwarding method uses a fixed-length per packet path ID and requires additional router storage proportional to the number of paths calculated. This is a significant improvement over the traditional source routing technique that requires variable length per packet path IDs, and routing storage overhead proportional to the number of paths calculated and their path length. The proposed forwarding method is a key component for efficient implementations of multipath routing.

Contribution 3: MPTCP, a multipath transport protocol that effectively utilizes available bandwidth on multiple paths.

The performance gains of using multiple paths depend on how effective end-hosts use the paths to increase their performance. The appropriate usage of these paths varies as applications vary. For example, an FTP application needs to use multiple paths to increase throughput, whereas a telnet application needs to use paths to decrease delay. The final contribution of this thesis is the development of a transport protocol, called MPTCP, which uses multiple paths to increase end-to-end throughput.

MPTCP stands for multipath TCP and is derived from the single path Transfer Control Protocol (TCP) [83]. MPTCP provides a reliable bit stream service and relieves applications of the burden of managing multiple paths. MPTCP operates by opening TCP connections on different paths and multiplexing data among them. The receiving MPTCP protocol coalesces data from the different connections to restore the original message stream. MPTCP performs congestion and flow control, both critical for MPTCP's high perfor-

RED HAT 1012
RED HAT V. CAS

mance.

Simulation results show that MPTCP is able to effectively use the additional resources offered by multiple paths, even under heavy network utilization levels. This protocol demonstrates that immediate end-to-end performance gains can be obtained from multipath networks.

1.2 Dissertation Overview

The rest of the dissertation is organized as follows. Chapter 2 presents the routing background necessary for the remainder of the thesis and discusses the differences between the dynamic metric and multipath routing models. Chapter 3 describes the hybrid-Scout routing algorithm, presents simulation results, and specifies the conditions under which hybrid-Scout outperforms traditional dynamic metric algorithms.

Chapters 4 through 9 address issues in the multipath routing model. Chapter 4 provides a formal introduction to multipath routing and presents the cost and benefit tradeoffs of multipath networks. Chapters 5 – 7 develop the components necessary for multipath routing to succeed. Chapter 5 develops appropriate path calculation algorithms that calculate quality paths between nodes. These paths are supported by the multipath forwarding methods given in Chapter 6. Finally, Chapter 7 describes a transport protocol, MPTCP, that effectively uses the multiple paths to maximize throughput.

Chapter 8 describes the implementation of two multipath routing algorithms. These two algorithms are then tested in Chapter 9. Through simulation, Chapter 9 evaluates the entire multipath routing model, from the performance of user protocols to the costs incurred by the two routing algorithms. The results conclude that multipath routing can be implemented efficiently and sufficient benefits obtained.

Finally, Chapter 10 describes the related work, and Chapter 11 summarizes the research contributions of this thesis and discusses future research directions.

RED HAT 1012
RED HAT V. CAS

Chapter 2

Background

This chapter provides the necessary background for this thesis by describing the basic Internet routing model and its current routing algorithms. The deficiencies in Internet routing motivate the need for dynamic metric and multipath routing in large-scale data networks.

The chapter begins by describing the two predominant routing algorithms, the Distance Vector and Link State routing algorithms, which are the basis of many Internet routing protocols. Moreover, many of the routing algorithms developed in this thesis are based on these two algorithms. Sections 2.1 and 2.2 describe the Distance Vector and Link State algorithms respectively. To place these algorithms in context, a brief description of the Internet is given in Section 2.3.

Section 2.4 then compares the Internet's static metric single path routing model with the dynamic metric single path and static metric multipath routing models. This comparison highlights the advantages and disadvantages of each model and reveals performance critical issues in dynamic metric and multipath routing models.

2.1 The Distance Vector Routing Algorithm

The first routing algorithm described is the Distance Vector routing algorithm (DV), a distributed, adaptive routing algorithm that computes shortest paths between all node pairs. Based on a centralized method known as the Bellman-Ford algorithm [22, 62], DV is the basis of many practical routing algorithms currently in use [7, 111, 133].

RED HAT 1012
RED HAT V. CAS

Basic Distance Vector Algorithm

A DV router computes a forwarding table which is used to forward packets on the best know path to their destination. Each entry in a DV forwarding table contains three elements: the destination address, the next-hop neighbor on the known shortest path to that destination, and the cost of the known path. Each router's forwarding table is continually updated by the distributed DV algorithm to indicate, via the next-hop entry, the shortest path between nodes. Routers update their forwarding tables by exchanging path information with their neighbors via *distance vector packets* (DVP). A DVP carries the identity of the originating router and a list of distance vectors of the form (*dst_addr*, *cost*), which are taken from the router's forwarding table. An DVP entry (d, x) sent by a router R indicates that R can reach destination d, and the cost of the path is x.

The description of the DV computation uses the following notation. Let C_d^r be the cost of the known least-cost path to destination d in r's forwarding table, N_d^r the next-hop neighbor on the path to d recorded in r's forwarding table, and c_{rs} the cost of the link from r to s. DV assumes that the path cost is additive and the cost of each link is positive. That is, the cost C of a path $(x_1, \ldots, x_n)$ is $C = c_{x_1x_2} + c_{x_2x_3} + \ldots + c_{x_{n-1}x_n}$ and $c_{x_ix_{i+1}} > 0$, $\forall 1 \leq i < n$. The forwarding table in each router r is initialized as follows:

$$C_r^r = 0;\ \forall s : s \neq r, C_s^r = \infty.$$

When a router r receives a DVP $((d, C_d^s), \ldots)$ from a neighbor s, r updates its forwarding table as follows: for all destinations d in s's DVP,

$$if\ (C_d^s + c_{rs} < C_d^r\ or\ N_d^r = s)\ then\ (C_d^r = C_d^s + c_{rs} \text{ and } N_d^r = s).$$

Notice that the cost added to C_d^s is c_{rs}, the cost from r to s. This addition correctly computes the path cost from r to d through s. Router r locally computes c_{rs} by observing its link status to neighbor s.

RED HAT 1012
RED HAT V. CAS

Routers exchange DVPs with their neighbors periodically or in response to link/node failures or recoveries. It can be proved that after a bounded number of DVP exchanges following a topology or link cost change, all routers' forwarding tables will contain values reflecting the shortest paths to all destinations [22, 157].

After DV forwarding tables converge, routers will then forward packets to their destinations on the calculated shortest paths. Because DV computes all pairs single shortest paths, tagging a packet with its destination address uniquely identifies the packet's path through the network. Data packets in DV are delivered to their destinations as follows: whenever a router receives a packet, it finds the forwarding table entry for the packet's destination and forwards the packet to the next-hop router listed in the table entry. This process of looking up the next-hop address and forwarding a packet to the proper neighbor continues until the packet reaches its destination.

Dynamic Metric Distance Vector

To calculate dynamic metric paths, a DV router continually updates the cost of each of its out-going links. When a router observes that the difference between a link's current cost and its last advertised cost exceeds a certain threshold, the router initiates (or triggers) a Distance Vector route computation. Because link costs may change very frequently, thresholding is used to dampen the number of path recomputations. Another mechanism often used to further dampen the rate of recomputations is to use hold-downs [74]. A hold-down limits the number of route computations a router can trigger within a certain time interval.

Given that the cost change in link l causes router R to initiate a route computation, the steps of recomputation are as follows:

1. In R's forwarding table, for every path p with l as its out-going link, R updates p's cost to reflect l's new link cost.

2. For each path updated, R sends a DVP containing these paths (and their new costs) to all out-going links but l.

RED HAT 1012
RED HAT V. CAS

3. Upon receiving a DVP update, router P performs the usual path updates. After updating its forwarding table, P sends DVPs containing path p if 1) p is newly acquired (as in the static metric case) or 2) the new cost of p exceeds p's old cost by a threshold.

After the DVP exchanges, the paths calculated between nodes will reflect l's new cost. Furthermore, the amount of dynamic metric computation is reduced by the use of different thresholds in the computation.

2.2 The Link State Routing Algorithm

The Link State (LS) algorithm is another routing algorithm on which many protocols are based. Example of these protocols are found in the Internet and in ATM networks [3, 110, 114].

The Basic Link State Algorithm

In the LS routing algorithm, every router periodically broadcasts (via flooding) its local connectivity. This information is flooded in a Link State packet (LSP) which consists of the router's ID, a list of its neighbors' IDs, and the cost of each connecting link. After all routers broadcast their LSP, every router knows the entire network topology. A router then performs a shortest-path spanning tree computation rooted at itself [54]. From the computation of this tree, the shortest paths to all destinations in the network are known. This information is then encoded in the router's forwarding table. A LS forwarding table is a list of tuples of the form (*dst_addr*, *next-hop*), where *dst_addr* is the address of the destination, and *next-hop* is the neighboring router on the shortest path to that destination.

Since the LS algorithm also computes single shortest paths between nodes, the usage of LS forwarding tables to deliver packets to their destinations is the same as DV.

RED HAT 1012
RED HAT V. CAS

Dynamic Metric Link State

Like DV, path recomputations in dynamic metric LS are also triggered by link cost changes. Dynamic metric LS routers continually update the link costs of their outgoing links and trigger path recomputation when the difference between the current and previously advertised link costs exceeds a threshold. Again, similar to DV, cost thresholding and hold-downs are used to limit the number LS path computations.

Upon detecting such link cost changes, an LS router broadcasts an LSP that contains the new link cost. When a router receives an LSP, it performs its usual shortest-path spanning tree computation that incorporates the new LSP information. Thus, after every router receives a dynamically triggered LSP and recomputes shortest paths, the paths provided by a dynamic metric LS routing algorithm will reflect the new cost change.

2.3 The Internet

The Internet is a global data network, and as of 1999, it consists of approximately 40 million hosts and 20,000 routers [136, 142]. The Internet is a best effort datagram network: units of data transmission are packetized, and each packet is individually and independently delivered (hop-by-hop) to its destination without any guarantees of success. That is, packets can be dropped in transit. In this section, we describe the current Internet routing architecture and show how LS and DV routing algorithms operate in and scale to large data networks.

In order for a node to send packets to another node, the sending node must specify the destination node. In the Internet today, every node is uniquely identified by an 32 bit IP address*. Thus for node A to send a packet to node B, A must designate the IP address of B as the packet's destination. This universal naming allows every node in the Internet to uniquely identify every other node.

*In the next version of IP, called IPv6, nodes will be identified by an 128 bit address [26].

RED HAT 1012
RED HAT V. CAS

Physically, the Internet is a collection of smaller inter-networks called *routing domains* (or *autonomous systems*). Each routing domain is responsible for routing packets within itself. That is, packets destined to hosts in a routing domain are routed by the domain's routing algorithm. An intra-domain routing algorithm refers to a routing algorithm that routes packets within a domain, and an inter-domain routing algorithm is responsible for routing packets between domains. Using this organization, packets sent from one domain to another are first routed out of the sending domain, then to the destination domain, and finally to the destination host by the destination domain's routing algorithm. Thus, logical Internet connectivity is ensured by the cooperation of inter- and intra-domain routing algorithms.

Because of the size of the Internet, two methods are used to make the network scalable: subnetting and hierarchical routing domains. Subnetting uses the IP addressing structure so that routers can route packets based on a set of hosts instead of individual hosts, and hierarchical routing domains reduce the size of routing domains. Both of these methods are described below.

Subnetting

Internet addresses are divided into *subnets*. A subnet is identified by some IP address prefix. Whenever an organization wants to connect to the Internet, it needs to obtain a unique subnet address or a set of unique subnet addresses. Furthermore, every host owned by the organization that connects to the Internet needs to have an IP address such that the IP prefix of the host address is the same as one of the subnet addresses obtained by the organization. The number of bits in the IP prefix is variable length [65, 129, 133], and the exact number of bits is not directly relevant to our discussion.

With subnet addressing, routers outside a subnet need to know and maintain only a single route (i.e. state) for all the hosts in the subnet. That is, routers outside a particular subnet do not need to maintain a route for every hosts in that subnet, but rather, these routers only need to maintain one route to the subnet; this route is used to forward packets

RED HAT 1012
RED HAT V. CAS

to every host in that subnet. Because several thousand hosts can have the same subnet address, subnetting allows substantial space savings in routers, making routing over the Internet more scalable.

However, subnetting alone is not enough to provide scalability because there are hundreds of thousands of subnet addresses in the Internet. To further reduce the amount of information that routers have to process and store, hierarchical routing domains are used.

Hierarchical Routing

Hierarchical routing allows a routing domain to contain subnet addresses as well as other routing domains (represented by a set of subnet addresses). Conceptually, routing is hierarchically structured such that at the lowest level, a routing algorithm routes packets to hosts with the same subnet address. At the second level, a routing algorithm routes packets among second-level routing domains using the subnet addresses the routing domains encompass. Similarly, at level i, a routing algorithm routes packets to the appropriate level i routing domain that contains the packet's destination subnet address. Because of the inclusive property of routing domains, a packet is routed to level $i + 1$ if the packet is destined for a subnet address that is not in level i. Eventually, routers in the highest level of the hierarchy, say level L, know about every subnet address and the level L routing domain that contains the address.

Thus a packet destined for a different subnet first travels up the domain hierarchy, until it reaches the routing domain that is the first common ancestor of both the source and destination routing domains. From this ancestor domain, the packet then travels down the hierarchy until it reaches the destination domain. The destination domain's routing algorithm then ensures the packet reaches its destination.

This hierarchical organization achieves scalability because a routing algorithm operating in a level i routing domain only needs to compute paths to nodes at that level. A node at level i could be a host (if $i = 1$) or a level $i - 1$ routing domain. If a node is a routing domain, then the node advertises the subnet addresses that it encompasses. With this hi-

RED HAT 1012
RED HAT V. CAS

erarchical structure, a routing algorithm at a particular level only has to compute paths to nodes at that level, thereby reducing the size of path recomputations that routers need to perform.

Conceptually, the Internet routing hierarchy can have many levels. However, in practice, the Internet routing is divided into only two levels, intra-domain (lower level) and inter-domain (higher level) routing levels.

Although the size of host addresses is different from domain addresses, a router's basic mechanisms for computing paths and forwarding packets are the same. Thus, the routing algorithms described in this thesis are applicable for all the levels of the routing hierarchy.

More information on Internet routing is available in standard network text books such as [93, 129, 154].

2.4 Comparison of Routing Models

As stated in Chapter 1, dynamic metric single path and static metric multipath routing models can offer higher network performance compared to current Internet routing algorithms that implement the static metric single path routing model. This section compares the three routing models and examines how their differences influence network performance.

Our comparison begins by examining routing algorithms under dynamic environments. This discussion applies to both single path and multipath routing models. Next, Sections 2.4.2 and 2.4.3 compare the performance differences among the three different routing models: Section 2.4.2 presents how network performance is influenced by each model's resource usage, and Section 2.4.3 compares the operation granularity of multipath and dynamic metric routing models. Operation granularity refers to the scale in which a routing algorithm computes its paths.

2.4.1 Static and Dynamic Metrics

A distributed routing algorithm (both single path and multipath) needs time t in order to converge on the computation of its paths. This convergence time depends on factors such

RED HAT 1012
RED HAT V. CAS

as the actual routing algorithm, network topology, link transmission speed, and router computation speed. For example, in the LS algorithm, convergence time depends on the time needed to detect topology changes, to broadcast topology information, and to recompute shortest paths.

When the network state changes, one expects that after time t, a routing algorithm will converge and calculate the appropriate paths based on the new network state. For example, assume that the convergence time for an LS algorithm is 5 seconds. Then 5 seconds after a link failure, the LS algorithm will recompute shortest paths such that the new paths do not use the failed link. In this case, the LS algorithm is correct because it calculates the expected paths (the shortest path between nodes) and does so within 5 seconds. However, the notion of correctness and convergence is less clear in scenarios where the network state changes faster than a routing algorithm's convergence time. This type of network scenario typically occurs in dynamic metric routing because link costs can change much more frequently than a routing algorithm's convergence time.[†]

In the previous example, assume that the network topology (or link costs) changes every second, then the LS algorithm can never catch up to the network changes and can never compute paths that reflect the actual shortest path in the current network. Rather, the paths that the LS algorithm computes will reflect the shortest paths in an out-dated network state, where the degree of out-datedness depends on the algorithm's convergence time. In other words, as the network changes, LS *tracks* the changes and computes paths according to its most up-to-date view of the network, which may be the actual network state sometime in the past. Note that system A tracks system B if whenever B changes to B', A also changes (say to A') such that the difference between the output of A' and B' is less than or equal to the difference between the output of A and B' [31]. Thus, a routing algorithm that tracks network changes will recompute its paths such that the newly computed paths better reflect the current shortest paths, compared to the previously computed paths.

[†]Although frequent network changes can also occur in static metric networks, these are usually networks under special conditions such as highly mobile, wireless networks [87, 127].

RED HAT 1012
RED HAT V. CAS

In network scenarios where a routing algorithm does not have an up-to-date view of the current network state[‡], one can still reason about the correctness of a routing algorithm. Recall in networks where rates of change are slower than convergence time, a routing algorithm is correct if it computes the expected path within its convergence time. Since a routing algorithm computes paths based on its current view of the network, such algorithms are correct if 1) their view of the network is *consistent* with the actual network and 2) the expected paths are computed based on that view. A router has a consistent view of the network if the router's knowledge of the current network state is the same as the actual state of the physical network.

In situations where a routing algorithm's network view is not consistent with the current network, a correctness criterion must ensure that the routing algorithm does not diverge in its shortest path computation. That is, the paths that a routing algorithm *actually* calculates should track the paths that the routing algorithm *would* calculate given it has the instantaneous network state. This property can be ensured by two criteria: 1) the algorithm's view of the network must track the actual state of the network. That is, the difference in the view of the network at time X and $X + \delta$ should reflect the actual network changes observed in the elapsed time δ. Of course, the amount of network updates during the elapsed time depends on the actual parameters of the routing algorithm (e.g. threshold and hold-down values). 2) Correct paths are calculated based on the algorithm's view of the network state. In the LS example, this means that given LS's view of the network, the shortest paths are calculated between nodes.

Assuming that dynamic metric versions of DV and LS are not guaranteed to have an up-to-date view of the network state, both algorithms are still correct according to the criteria given above. In both algorithms, shortest paths are computed based on each algorithm's

[‡] A routing algorithm might not have an up-to-date network view either because the network state changes too frequently or because the algorithm does not attempt to obtain the exact, moment-to-moment state of the network due to efficiency issues. Mechanisms such as thresholding and hold-downs allow routing algorithms to reduce routing cost at the expense of less accurate network view.

RED HAT 1012
RED HAT V. CAS

current view of the network, and network views are continuely updated via triggered path recomputations. In LS, the network view is updated by topology broadcasts, and each broadcast partially updates the network view. Similarly in DV, the network view is updated by messages received from neighboring routers that contain path information that use current network link costs.

In practice, this implies that the accuracy with which a dynamic metric routing algorithm tracks physical link costs directly affects the quality of the paths calculated. The closer the tracking, the closer the calculated paths reflect the current state of the network; however the cost of tracking the network state comes at a cost. Chapter 3 addresses this issue and shows that achieving good network performance involves a tradeoff between the cost of tracking the physical network and the quality of the paths calculated.

2.4.2 Resource Usage

This section compares the three routing models in terms of network resource consumption and shows how their differences affect network performance. To understand these issues, one needs to determine 1) what network resources are, 2) how these resources are consumed, and 3) how network performance is defined. These terms are defined below.

First, there are two types of network resources, router and link resources: router resources refer to router CPU cycles and memory, and link resources refer to link bandwidth. Second, these network resources are consumed by two types of messages: data messages and routing messages. Routing messages are sent by routers to compute and provide paths between nodes, and data messages are sent by applications to other applications. Both types of messages consume link resources as they traverse the network and router resources because routers receiving these messages have to process and/or forward them. Third, network performance is determined by application network demands. Examples of different network demands are high throughput, low delay, and high probability of satisfying QoS requests[§]. Although network performance is application specific, in general, performance

[§]Quality of Service (*QoS*) specifies a level of network service, typically expressed as a combination of

RED HAT 1012
RED HAT V. CAS

can be characterized by the speed in which data messages are delivered to their destinations: the faster data messages are delivered from their sources to their destinations, the higher the network performance. We shall use this general measure of network performance for the remainder of the discussion.

Given these definitions, the following subsections analyze how the three routing models trade off between network performance and resource consumption of data and routing messages.

Data Messages

This subsection examines how data messages use network resources as they travel to their destinations. In static metric single path routing, the network resources used by data messages between a node pair are fixed given a particular topology: the paths provided are based on static link costs and do not change with traffic conditions. The drawback of this routing model is that network performance is dependent on traffic; therefore depending on traffic patterns, the static metric shortest path might not provide good network performance.

In dynamic metric routing, the cost of a link considers the traffic on the link. In this model, a link's cost changes dynamically – a link's cost increases if the amount of traffic on the link increases. Thus, in dynamic metric routing, the network resources used between a node pair change (via path recomputation) in order to minimize dynamic path costs and to increase performance. In essence, with dynamic metric routing, data messages travel different paths (i.e. use different network resources) depending on network traffic patterns.

In the multipath routing model, the routing algorithm provides multiple paths between nodes and allows the sending nodes to choose how to send data using the provided paths.

---

delay bounds and capacity requirements [9, 43, 60, 71, 95]. For example, a video conferencing application that transmit real-time video and audio may need paths with 1Mb/s bandwidth and a delay bound of 10ms. In this scenario, the video conferencing application needs to reserve this QoS specification on the path to its destination. Notice that this request may be rejected by the network due to resource availability. However, once reserved, the network is expected to provide the requested path performance to the application.

RED HAT 1012
RED HAT V. CAS

With respect to network resource usage, the resources a message uses to its destination depend on the paths calculated by the routing algorithm and on the end-host sending the message. The paths that data messages *can* travel depend on the paths calculated by the multipath routing algorithm. However, the path that data packets actual travel depends on the end-host sending the packets. Thus, in multipath routing, end-hosts can increase their performance given that they are able to gauge the performance of their available paths and send data on paths that best maximize their performance.

In summary, both dynamic metric and multipath models offer higher performance than static metric shortest routing by delivering data on paths which consider observed network traffic. In dynamic metric routing, the routing algorithm observes network traffic and recomputes its paths to optimize a given performance metric. The data messages then automatically traverse the recomputed paths. With multipath routing, the multipath routing algorithm simply computes multiple paths between nodes. It is the responsibility of the end-hosts to observe the performance of the computed paths and send data on paths that best increases their performance. This difference in data resource usage affect the two models' cost-performance tradeoff. The tradeoff difference is discussed further in subsequent sections.

Routing Messages

Routing messages are the other type of message that consume network resources. This section considers the tradeoff between their resource consumption and network performance.

With respect to this tradeoff, the static metric single shortest path routing is at one end of the spectrum that minimizes routing resources. This model uses the least amount of resources among the three models because it computes only one path between a node pair, and these paths do not change unless the network topology changes. In contrast, dynamic metric and multipath models use more routing costs in the attempt to provide higher network performance.

In the dynamic metric routing model, routing messages are used to track the network

RED HAT 1012
RED HAT V. CAS

state (e.g. link costs). Thus, if an algorithm uses more routing messages, it has a more accurate view of the physical network, resulting in higher path quality because the computed least-cost paths more closely reflect the actual least-cost paths in the physical network. This higher path quality, in turn, increases performance because better paths increase the speed that data messages reach their destinations (given that link metrics reflect packet transmission time). However, because routing messages also consume network resources, which affects network performance, dynamic metric algorithms must balance between the accuracy of their network state information and the potential performance degradation of obtaining the information.

Multipath routing algorithms also incur extra routing overhead in exchange for potentially higher network performance. Extra overhead is incurred through calculating and maintaining multiple paths between nodes. Thus, multipath routing algorithms must balance between the resources needed to compute and maintain paths versus their potential performance gains.

To summarize, compared to static metric single shortest path routing, both dynamic metric and multipath routing require more routing message resources. These extra resources allow both models to obtain higher network performance. However, there is a fundamental difference between dynamic metric and multipath cost-performance tradeoff. In dynamic metric routing, the current least-cost paths between nodes are determined by the routing algorithm. Therefore, changing paths between two nodes requires routing intervention, which entails routing costs. In multipath routing, however, the cost of calculating multiple paths is incurred *once* for a particular network topology, and subsequent path changes to avoid congestion are done by end-hosts and thus do not require routing intervention or routing overhead. For this reason, the multipath routing model is inherently more efficient than dynamic metric routing at making the routing cost versus network performance tradeoff. We address this issue in more detail next.

RED HAT 1012
RED HAT V. CAS

2.4.3 Granularity

As shown in the previous section, both multipath and dynamic metric routing increase network performance by using network resources that consider network traffic. However, the difference between the two models is that end-hosts in multipath routing choose which paths to use, whereas nodes in dynamic metric routing do not. This section explains how this difference serves to distinguish the two routing models.

At a very abstract level, one can imagine that a dynamic metric routing algorithm can "simulate" a multipath routing algorithm by allowing nodes to specify the desired path before sending a packet and then compute the path fast enough that the routing algorithm provides an illusion that there are multiple paths between nodes. This section shows that this cannot be realized in practice. The reason points to the fundamental difference between the two models – the granularity in which the two models operate. The difference can be categorized into *time* and *path* granularities.

Time granularity refers to the time scale in which nodes can send data on different paths. In a multipath model, nodes have the freedom to send data on any of their available paths. For example, a node can send consecutive messages on different paths in a round-robin fashion. That is, the time granularity of path switching is on the message level (or on the order of microseconds). However, in dynamic metric routing, switching paths between nodes requires global path recomputation which is on a much larger time granularity, typically on the order of seconds. The reason for the larger time granularity is due to efficiency: computing paths on the microsecond granularity would consume a prohibitively high amount of network resources, which can significantly reduce network performance. Notice that in multipath routing, switching between paths does not incur any routing overhead.

Path granularity refers to the scale of switching paths. In a multipath setting, when a node decides to send data on different paths, the decision is made by the sending node alone and affects only that node's performance. That is, path switching is on an end-to-end granularity. In dynamic metric routing, however, path recalculation is done on a network gran-

RED HAT 1012
RED HAT V. CAS

ularity because path recomputation entails global calculation. This computation involves potentially every router in the network and potentially affects paths of other connections. Because of this global granularity, it is thus infeasible to allow nodes in a dynamic metric environment to arbitrarily switch paths and request global path computations. Notice that the end-to-end path granularity in the multipath model naturally allows nodes to freely switch paths.

Because multipath and dynamic metric models offer different operating granularities, in practice, the two models are very different and require different routing algorithms to implement each model.

2.4.4 Routing Model Summary

In summary, both dynamic metric single path and static metric multipath models trade off resource consumption for increased network performance by delivering data on paths that consider the dynamics of network traffic. In the dynamic metric shortest path model, resources are used to more accurately track the network state, resulting in better calculated paths. Multipath routing, on the other hand, uses additional resources to compute and maintain multiple paths, thereby allowing end-hosts to achieve higher network performance.

Analysis of this cost–performance tradeoff reveals that multipath routing is intrinsically more efficient than dynamic metric routing. This is derived from the observation that to provide increased end-to-end performance, a multipath routing algorithm needs to compute its paths only once (until the network topology changes), while a dynamic metric algorithm needs to continually recompute paths to maintain path quality.

Although dynamic metric algorithms make this tradeoff less efficiently, an advantage of this model is that it only requires changes in router software and minimal end-host changes. Since dynamic metric routing provides one path between a node pair, current end-host protocols and softwares need little change, if any, to obtain performance benefits. In contrast, multipath routing requires updates to both router and host software in order to reap multipath performance benefits.

RED HAT 1012
RED HAT V. CAS

As this section shows, the decision of which routing model to implement requires evaluating tradeoffs at various levels. Analysis of such tradeoffs continues throughout the remainder of this thesis. The following chapters carefully consider the cost–performance tradeoffs when developing algorithms that implement dynamic metric and multipath routing models.

RED HAT 1012
RED HAT V. CAS

Chapter 3

Dynamic Metric Routing

This chapter presents a novel routing algorithm that implements the dynamic metric shortest path routing model. Dynamic metric routing has been implemented on the ARPANET (the precursor of the Internet) and substantial improvements in network performance were observed: Khanna and Zinky [94] showed that dynamic metric routing has a significant impact on the ARPANET's performance, where performance is measured in packet drops. This dynamic cost metric, called the *revised ARPANET routing metric*, also reduced routing instabilities and oscillations compared to the previous dynamic metric based purely on link delays. Despite their positive results, dynamic metrics are not widely used in today's Internet. The technical reasons are that 1) the amount of routing updates with dynamic metrics are hard to control *a priori* because routing updates are dependent on network traffic, and 2) routing using dynamic metrics can cause routing oscillations, though Khanna and Zinky's dynamic metric lessens the problem.

The dynamic metric routing algorithm presented in this chapter to a large extent, overcomes the limitations cited above. The algorithm uses *Scout*, a destination-initiated shortest path computation technique. A destination node using the Scout algorithm initiates a path computation from every node in the network to itself at periodic intervals. The period between route recomputations is controlled by the initiating node. The routing overhead of the algorithm is a function of this period and is independent of the dynamic conditions in the network. Therefore, Scout has the property that its routing overhead is predictable and under the control of the initiating nodes.

Since each initiating node takes charge of its route recomputations, the updating of routes to each initiating node is uncorrelated. In contrast, traditional dynamic metric al-

RED HAT 1012
RED HAT V. CAS

gorithms based on Link State (LS) and Distance Vector (DV) recompute paths between all nodes quasi-simultaneously. This quasi-simultaneous computation has the unfortunate tendency of shifting *congestion causing traffic* from a currently congested area to a currently uncongested one which then becomes the next congested area, triggering route recomputations over again [163]. Because route recomputations in Scout are independently controlled by the destinations, route updates to different destinations are uncorrelated and usually staggered in time, thus allowing congested traffic to be split, significantly reducing route oscillations caused by shifting all traffic in a congested zone at the same time.

However, this flexibility and autonomy in route recomputation comes at a price. Scout is not as efficient as algorithms such as DV in computing all-pairs shortest paths. This is because Scout destinations independently compute paths to themselves and do not share path computations with other nodes; thus, using Scout to compute dynamic metric paths to all nodes is inefficient.

To obtain the benefits of dynamic metric Scout while maintaining low routing costs, Scout is integrated with the traditional LS and DV algorithms. In this hybrid algorithm, the LS or DV component operates as usual and calculates static metric shortest paths. This provides good baseline paths between nodes. The Scout component then computes dynamic metric paths to a small, selected number of "hot" destinations. Analysis of Internet packet traces show that a high percentage of Internet traffic is destined to a small percentage of "hot" destination networks (see Figure 3.1). The main idea behind the hybrid is that if a small number of destinations receive the majority of the traffic, then making sure that their traffic do not experience congestion (by calculating dynamic metric paths to these destinations) will reduce the probability that the remaining traffic experiences congestion.

By using Scout with dynamic metrics to improve path quality to hot destinations and relying on traditional static metric algorithms to maintain paths to other nodes, the hybrid algorithm reaps the benefits of dynamic metric routing without paying a high, unpredictable routing overhead and without incurring significant routing instabilities. Moreover, since static metric DV and LS algorithms efficiently calculate all-pairs shortest paths and Scout

RED HAT 1012
RED HAT V. CAS

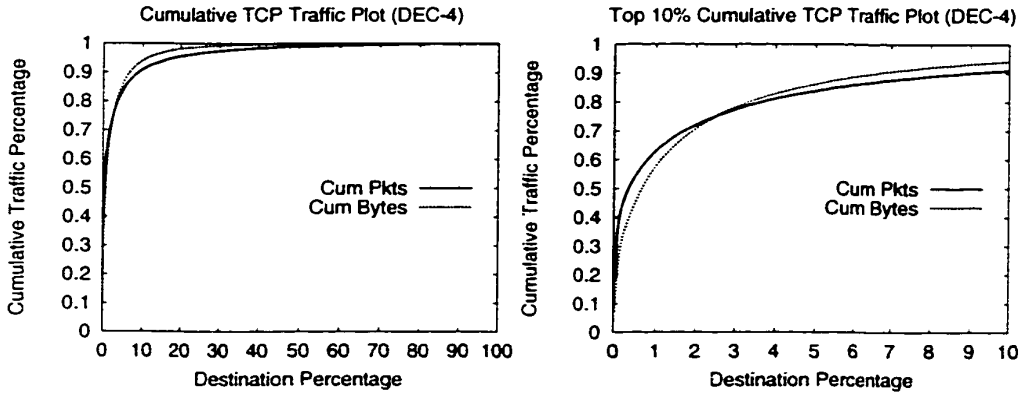


Figure 3.1 : Traffic locality in dec-pkt-4 TCP packet traces from *http://ita.ee.lbl.gov/html/contrib/*. The graph shows the cumulative distribution of network traffic to destinations. The trace contained an hour's worth of wide-area traffic from DEC's corporate intra-net to the rest of the Internet. The x-axis shows the destination percentage sorted by traffic frequency, and the y-axis shows the cumulative distribution. The left graph shows the cumulative distribution of traffic to all destinations, and the right graph shows distribution to the top 10% of destinations that receive traffic. The distributions of traffic to sources exhibit the same distributions. The locality shown in this trace is typical of other traces analyzed.

efficiently calculates paths selectively, the hybrid algorithm combines the strength of the different algorithms.

The rest of this chapter is organized as follows. Section 3.1 presents the hybrid approach. The section first describes Scout, a selective route recomputation algorithm that takes advantage of network traffic locality. Scout description is followed by two sections that describe Scout integration with DV and LS routing algorithms. Section 3.2 presents experimental data showing the behavior of the different dynamic metric algorithms. The results of the simulation show that hybrid Scout outperforms traditional dynamic metric algorithms using significantly lower routing costs. Finally, Section 3.3 concludes this chapter.

RED HAT 1012
RED HAT V. CAS

3.1 The Hybrid Approach

One of the key ideas of the hybrid approach is that routes based on dynamic metrics are computed only for a minority of destinations, namely destinations that receive a majority of the traffic. Due to the high destination locality observed in network traffic, being able to calculate dynamic metric paths to these *hot destinations* has several advantages. First, one hopes that calculating paths to a few destinations is cheaper than calculating paths to all destinations. Second, not calculating shortest paths to all destinations simultaneously reduces the route heredity problem [94] and therefore reduces route oscillation. In addition, because these destinations receive a high percentage of routing traffic (say 50+%), rerouting these paths enables the algorithm to shift enough traffic to alleviate areas of congestion.

The solution presented here for dynamic metric routing is based on two key concepts. The first is the Scout routing algorithm that is able to selectively update paths to individual destinations. The second key idea is to integrate Scout with traditional DV and LS algorithms. This hybrid provides base static routes to a majority of destinations using DV/LS and provides dynamic metric routes to selected destinations using Scout.

Scout is a host-initiated, selective route calculation algorithm that calculates least-cost paths to individual nodes (or networks). Host-initiated means that the path calculation process to a node (or network) is initiated by that node (or gateway router to the network). Scout is selective because only paths to that destination are affected, and it is efficient at computing routes to individual destinations. However, Scout does not aggregate path computation, and therefore is not as efficient as LS and DV when computing paths to all nodes. This deficiency motivated the integration of Scout with LS and DV.

The hybrid of Scout with DV and LS routing algorithms uses the traditional routing algorithm to calculate paths between nodes using static metrics, while Scout continually refines paths to selected nodes using dynamic metrics.

The remainder of this section is dedicated to describing the hybrid Scout algorithm. The next section presents a brief description of the basic Scout algorithm and shows how it computes least-cost paths between nodes. The integration of Scout with LS and DV follows

RED HAT 1012
RED HAT V. CAS

in Sections 3.1.3 and 3.1.4. Finally, Section 3.1.5 summarizes the hybrid-Scout LS and DV algorithms.

3.1.1 The Scout Algorithm

This subsection briefly describes the Scout routing algorithm. Appendix A contains a full description of Scout, including a proof of its correctness and convergence. The Scout algorithm is described in a static metric network environment; Section 3.1.2 discusses the algorithmic implications of using dynamic metrics.

In the Scout routing algorithm, each router maintains a forwarding table indicating for each destination, the next-hop neighbor on the known least-cost path to that destination. The forwarding tables are updated by Scout and are used in the same way as in LS and DV.

Scout's basic mechanism of route computation is message *flooding**. With Scout, destinations periodically flood a *Scout* message throughout the network. Let R be the node initiating the Scout message. The period between two consecutive floodings of Scout messages from R is called the *broadcast interval* (BI). A Scout $[R, C_R, x]$ contains the originating node's address, R, and the cost C_R to reach R, initially zero, and an increasing sequence number x. When a node P receives a Scout message $[R, C_R, x]$ from its neighbor Q, P first checks whether the sequence number x is valid. If not, the Scout is discarded. Otherwise, P modifies the Scout's cost C_R to include the cost of sending a message from P to Q, $C'_R = C_R + Cost(P \rightarrow Q)$. C'_R represents the cost a message will take if it is sent by P via Q to R.

In the first broadcast interval (i.e. when P has no record of receiving a Scout from R), P forwards the Scout $[R, C'_R, x]$ immediately after receiving the first Scout message,

*Flooding is a standard method of broadcasting information in a point-to-point network. The flooding method operates as follows: whenever a node receives a flood message, it remembers the message and sends the message to all neighbors except the neighbor that it received the message from. If a node receives a flood message it has seen, the message is discarded. Thus, a flood message terminates after every node receives the message.

RED HAT 1012
RED HAT V. CAS

initiated by R, to all neighbors except Q, the neighbor which sent P the Scout. Node P might receive more of R's Scouts in the same BI, indicating different paths and path costs to R. P remembers only the least-cost Scout to R and adjusts its forwarding table to reflect the best known path, but P does not forward these Scout messages. For each node, the next-hop to R is always the neighbor that provided the least-cost Scout in the current BI. In the next and subsequent broadcast intervals of R, P waits to receive a Scout message from its *designated neighbor* before flooding. We define node P's *designated neighbor* to node R as the neighbor of P that provided the least-cost Scout to R in the previous broadcast interval. When P receives the Scout from this designated neighbor, P forwards the least-cost Scout received in the current BI (i.e. with the current sequence number) to R to all neighbors except the one from which the least-cost Scout was received.

In the presence of node or link failures, a node P might never receive a Scout from its designated neighbor because the designated neighbor may no longer be connected. This is taken into account by requiring P to flood the first Scout message from R if P did not flood any of R's Scout messages in the previous BI. In other words, if P waited for a Scout from its designated neighbor Q in the previous BI but never received a Scout, instead of waiting for Q, or any other neighbor in the current round, P immediately floods the first Scout it sees in the current broadcast interval.

In this case, P is not allowed to wait for a neighbor in the current BI, in particular, the designated neighbor, because if there are multiple failures, waiting for the best information might cause cascading waits. Our decision was motivated by the observation that propagating more recent, perhaps sub-optimal, information is more useful than trying to wait for the best information which entails the risk of not propagating any information at all.

As an example, consider node A sending Scouts in its first broadcast interval (sequence number 0). A trace of how Scouts propagate through the network is shown in Figure 3.2. Node A sends a Scout of the form $[A, 0, 0]$, denoting the Scout originates from A, with cost 0, sequence number 0.

In this example, node A's Scout forwarded by B reaches D before the one forwarded by

RED HAT 1012
RED HAT V. CAS

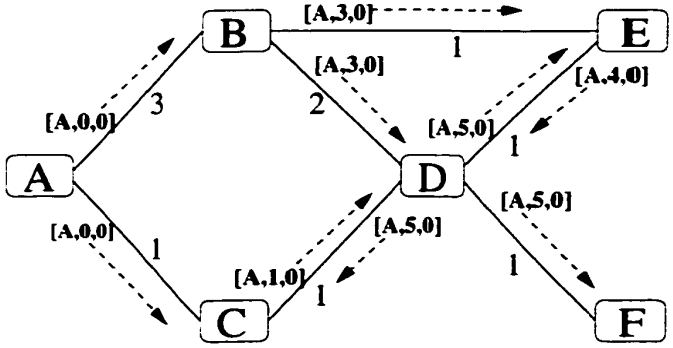


Figure 3.2 : A Scout Example. Node A Sends Scouts for the first time. In this network, boxes denote routers and edges links. The number beside each link represents the cost of that link.

C. Since D has no previous information of A's Scout, D sends the Scout $[A, 5, 0]$ to all its neighbors, after adjusting the Scout cost. Notice that D learns the optimal path to A when it receives the Scout $[A, 1, 0]$ from C. However, D does not propagate this information in the current BI. Instead, C becomes D's designated neighbor (and next hop to A). In the first BI, the only node that did not get the optimal cost to A is node F, even though it has the optimal route. That is, because of hop-by-hop forwarding, F's packets destined to A will traverse the path (F, D, C, A) with cost 3 even though F thinks it's optimal path cost is 6.

In the next BI, D will wait for C's Scout before sending its Scouts. The trace is given in Figure 3.3. Notice that D sends the Scout from C. The algorithm converges in two rounds and every node has the least-cost path to A.

There are five distinctive features of the Scout routing algorithm. First, the number of Scouts forwarded in one BI is $O(L)$, where L is the number of links[†]. Second, in a steady state network, the worst-case number of BI's needed for Scout to converge is proportional to the longest calculated cost path (with respect to the originating node). However, this worst case requires pathological conditions. In practice, Scout's convergence is much better.

[†]It is not exactly L because it may happen that two neighboring nodes send Scouts to each other simultaneously. The number of Scouts is between L and $2L$.

RED HAT 1012
RED HAT V. CAS

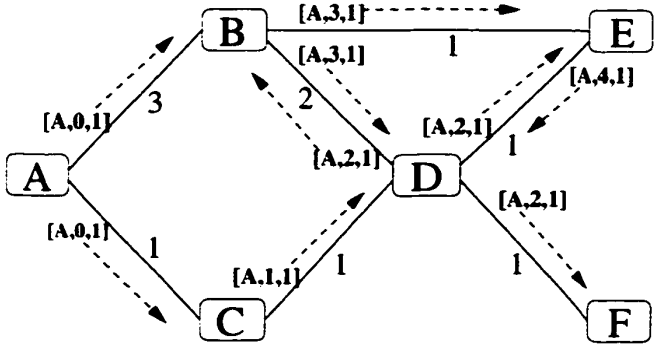


Figure 3.3 : A Scout Example. A sends Scout in the second BI

Simulations were conducted on networks with more than 100 nodes which showed that Scout always converged in no more than three BI's. Moreover, after a link failure, an alternate (not necessarily least-cost) path is guaranteed to be computed after at most two BI's, if one exists.

Third, Scouts are small and fixed size and can therefore be piggybacked on a hop-by-hop basis onto data packets, largely defraying their cost to the network. Fourth, Scout's route updates are completely controlled by the Scout generating node and not by network changes; therefore Scout routing costs are easily controlled. Fifth, Scouts require very little router computation. This is in contrast to LS and DV where a routing update may affect entire forwarding tables.

3.1.2 Dynamic Metric Scout

The Scout routing algorithm can also compute paths using dynamic metrics. That is, Scout can calculate paths to destinations when link costs change dynamically. The implementation of dynamic metric Scout is straightforward. Routers maintain each out-going link's dynamic cost as in dynamic metric LS and DV, and upon receiving a Scout from link l, the router adds l's current dynamic cost to the Scout cost and proceeds according to the usual Scout algorithm.

Given that the number of BI's for Scout to converge is proportional the longest cal-

RED HAT 1012
RED HAT V. CAS

culated path, it may be that the rate of link cost changes in a dynamic metric network is higher than the Scout convergence rate. In this case, Scout might never converge to the actual dynamic metric shortest path to a particular destination. As stated in Section 2.4.1, the correctness of dynamic metric Scout in this scenario depends on 1) whether Scout's view of the network tracks the physical network and 2) whether paths are correctly calculated according to Scout's current view.

Similar to DV, a Scout router updates its view of the network by receiving Scout messages from its neighbors. Scout messages update the network view because whenever a router forwards a Scout, the most up-to-date link costs are added to the Scout. Thus, the network view to a destination is tracked as the destination sends Scouts on every BI.

With respect to the second condition, the dynamic metric Scout algorithm also calculates the appropriate path given its current network view: a router always stores current BI's least-cost Scout in its forwarding table, and routers always forward the current BI's least-cost Scout as dictated by the Scout algorithm. Because Scout is an iterative algorithm, Scout might not calculate the shortest path to a destination after one Scout broadcast. Therefore, although a Scout broadcast from destination d does not guarantee that every router after the broadcast will have the shortest path to d, the calculation process is still correct because Scout update and forwarding procedures guarantee reductions in path costs and produce shortest paths in a steady state network.

Like dynamic metric LS and DV, the dynamic metric Scout algorithm trades off between routing costs and the accuracy that the calculated paths approximate actual least-cost paths. For example, in dynamic metric LS and DV, the smaller the threshold and hold-downs, the more closely the calculated path track actual least-cost paths. Smaller thresholds and hold-down values allow more frequent routing updates; thus the routing algorithm's view of the network more closely tracks the actual network. Similarly, the smaller the Scout BI's, the better Scout paths track least-cost paths because smaller elapsed time between consecutive BI's reduces the difference between Scout's view of link costs and the actual costs.

RED HAT 1012
RED HAT V. CAS

The cost and performance of a dynamic metric routing algorithm can be measured by how much routing resources (in terms of routing messages, router CPU, etc.) the algorithm uses in order to provide a certain level of path quality. In our experiments, the cost of a link is a function of the link's queue length, which directly reflects link delay. Using this link metric, path quality is measured by the time a packet reaches its destination. Therefore, an algorithm that provides a lower overall network packet delay provides better performance (or tracks network changes more closely) than algorithms that provide higher delays. An ideal dynamic metric routing algorithm would provide minimum overall network packet delay. Comparisons of routing cost and performance (in terms of packet delay) for hybrid Scout and dynamic metric LS/DV algorithms are given in Section 3.2.

3.1.3 Scout-DV Hybrid

The Scout-DV hybrid algorithm combines the Distance Vector (DV) routing algorithm with Scout. This section describes the integration of the two algorithms and justifies the behavior of the resulting hybrid routing algorithm.

Scout-DV Algorithm

Scout-DV algorithm has a DV component and a Scout component. The DV component is responsible for computing the least-cost paths using static link costs and propagates link failures and recoveries in the usual way. The Scout algorithm is integrated with only a few small changes described below. The Scout component directly modifies the DV forwarding tables (the next-hop and cost fields) and uses dynamic link metrics.

Because the DV component of the hybrid algorithm computes static shortest paths between all destinations, the Scout component of the algorithm can take advantage of this. The only modification to Scout is the following:

- If the Scout algorithm does not have any information on the designated neighbor, the default is to use the next-hop neighbor computed by the DV algorithm as the designated neighbor.

RED HAT 1012
RED HAT V. CAS

This feature is used in two places: 1) when a node generates its first Scout message and 2) during the first Scout broadcast after a link failure/recovery (see below).

The two modifications to the DV algorithm are:

- When a router detects a link/node recovery, it exchanges forwarding tables with neighboring routers in the usual DV manner, propagating the Scout paths as if they were computed by DV. The receiving router performs the standard operations on the received DVP and recomputes the shortest paths for both Scout and non-Scout sending destinations.

- When a router detects a link/node failure, it follows the standard DV procedure and sends to all its neighbors a DVP containing infinity cost for all destinations that use the failed link (including Scout sending destinations). This process nullifies all DV and Scout paths that use the failed link. Again, the standard DV recomputation mechanism will compute static metric shortest paths (if they exist) to all affected destinations.

  To ensure that a straggling Scout message from the previous BI does not re-advertise a path through the failed link, the Scout-DV hybrid algorithm increments the Scout sequence counter for Scout calculating destinations that were affected by the failure. Notice this does not alter the Scout algorithm behavior in the next BI because the sequence number will then be current.

The above modifications to Scout and DV algorithms serve to better Scout's approximation of the least-cost path. The Scout modification provides an educated guess for a default designated neighbor: lacking any dynamic metric path information, the static metric shortest path is the most likely the shortest dynamic metric path. The DV modifications, on the other hand, ensure that the Scout paths affected by a component failure/recovery become *valid* in at least the same speed as affected DV paths. A *valid path* is defined as a path that connects the source of the path to the destination. In a distributive routing environment, this means that the forwarding table of every intermediate router on a valid path has, as its

RED HAT 1012
RED HAT V. CAS

next-hop field to the path's destination, the next router on the path and that every link/router along the path is operational. In other words, packets are forwarded to their destinations only along valid paths.

Notice in the hybrid algorithm, the speed of convergence for paths to non-selected destinations are the same as those in traditional DV algorithms (static metric DV). This is because only the DV component of the hybrid algorithm calculates paths to non-selected destinations. Therefore, paths to these destinations are updated by the DV component in exactly the same manner as ones updated by traditional DV algorithms; thus paths to non-selected destinations in hybrid-Scout must have the same convergence guarantees as destinations in traditional DV algorithms.

The next section shows that the modifications to the DV component in the presence of component failures/recoveries provide connectivity to Scout generating nodes.

Scout-DV Interactions

In the hybrid Scout-DV algorithm, the DV and Scout components interact in order to speed Scout's least-cost path computation and to speed the establishment of valid paths after link/node failures. The purpose of this section is to show that the DV-Scout interactions described above result in valid paths. Notice that the hybrid algorithm is still correct even if the two components are completely decoupled: both DV and Scout are capable of calculating shortest paths without help from the other.

To show that valid paths are computed as a result of Scout-DV interactions, a distinction is made between destinations that generate Scouts and destinations that do not. For those that do not, their paths are updated only by the DV component, and therefore the validity of their paths is the same as in the pure DV algorithm. The correctness of DV has been proven in [22,62].

With respect to destinations updated by Scout, their paths are also updated by the DV component. Only for these destinations do we need to verify that valid paths are computed. Because the Scout and DV components interact only during certain events, we need to

RED HAT 1012
RED HAT V. CAS

ensure that after those events (or after the interactions), valid paths are provided by the hybrid algorithm. The list below enumerates the points of Scout-DV interactions and the conditions that ensure the calculation of valid paths.

1. When Scout does not have a designated neighbor, Scout's usage of DV's next-hop neighbor as the default designated neighbor does not cause Scout to diverge in its path computation.

2. After a link failure, the DV component nullifies all Scout calculated paths that traverse the failed link. Moreover, DV calculates alternate paths and these paths are valid.

3. After link recovery, the DV component calculates valid paths.

Justification of 1: In the pure Scout algorithm, if a router receives a Scout which it does not have any previous information, the router will treat the neighbor that sent the first Scout as its designated neighbor. With the DV optimization, the hybrid Scout will use the DV's next-hop neighbor as its designated neighbor. Notice that in the subsequent BI, the Scout component will use the designated neighbor it computed from its previous BI. Thus this optimization can only affect Scout behavior for the first BI; therefore the optimization does not affect Scout correctness (i.e. its approximation of the least-cost path) in calculating shortest paths to Scout generating nodes.

Justification of 2 and 3: In DV, whenever a router detects that a link has failed, it nullifies forwarding table entries to destinations whose next-hop is through the failed link. In addition, the router sends a DVP to all its neighbors indicating that it can no longer reach the list of nullified destinations. The sending of DVP's then initiates a network wide path cancellation and recalculation process.

The complication with Scout-DV is that the DV component nullifies Scout calculated paths and later recomputes these paths. The concern is whether this interaction causes the calculation of invalid paths to Scout destinations.

RED HAT 1012
RED HAT V. CAS

To show that valid paths are computed as a result of interactions 2 and 3, we must show that after the DV recalculation process, the collective Scout-DV forwarding tables reflect, for every Scout destination affected by the topology change, a sequence of valid next-hops that leads to the destination.

The crux of this justification is that Scout paths have exactly the same properties as DV paths. Therefore, when Scout paths are nullified and recomputed by the DV component, these paths will have the same resulting properties as paths calculated by DV. Thus the validity of Scout paths that are (re)calculated by the DV component is reduced to the validity of DV paths. Given that the correctness of DV has already been established, reducing Scout paths to DV paths shows that interactions 2 and 3 will produce valid paths to Scout generating nodes.

The justification of interactions 2 and 3 proceeds as follows: first, we state the necessary and sufficient conditions of DV paths. Second, we show that Scout paths have the same properties as DV paths. The validity of DV-modified Scout paths then directly follows.

From the Bellman equations [22], the three necessary and sufficient conditions of a path $(x_0, \ldots, x_n)$ calculated by DV are the following:

1. Node x_i's forwarding table entry has x_{i+1} as its next-hop to x_n, $0 \leq i < n$.

2. The cost x_{i+1} has to x_n is less than x_i's cost to x_n, $0 \leq i < n$.

3. The information of this path flowed from x_n to x_0 via the reverse path $(x_n, \ldots, x_0)$.

The Scout calculated paths also have these three properties: 1) if x_i calculates the least-cost path to x_n as $(x_i, \ldots, x_n)$, then x_i's forwarding table entry for x_n has x_{i+1} as its next-hop. This is from the basic Scout update rule. 2) Scout's costs are additive, satisfying the second property. 3) Like DV, Scout's path information flows from the destination in the reverse direction of the calculated path. This follows from the Scout flooding mechanism.

Because Scout's paths look exactly like DV computed paths, whenever the DV component updates router forwarding tables, it cannot distinguish the paths calculated by Scout from the ones calculated by the DV component itself. Thus Scout calculated paths will be

RED HAT 1012
RED HAT V. CAS

altered in the same way as DV calculated paths and will have the same DV path properties after link/node failure or recovery. That is, the DV component is guaranteed to nullify all invalid Scout paths after a link failure and recompute valid paths to Scout destinations on link recoveries. QED.

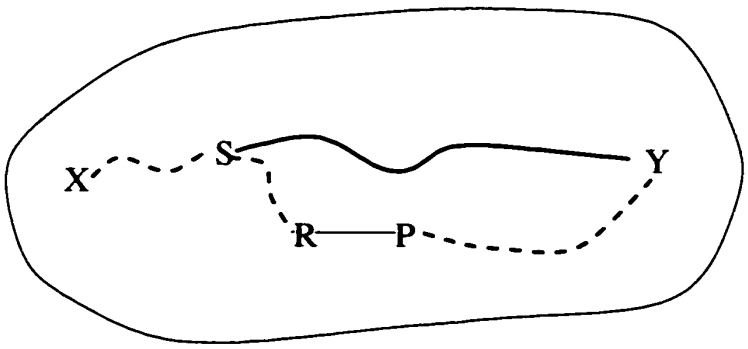


Figure 3.4 : An example of a path that the DV component calculates to a Scout generating destination after the failure of link (R, P).

Figure 3.4 shows an example of a Scout path invalidated by a link failure and subsequently recomputed by the DV component. In this figure, the dotted curve between nodes X and Y denotes the path node Y has to node X. Here, X is a Scout generating node. For this example, assume the link between R and P fails and that the DV component detects the failure before X generates another Scout[‡]. According to the hybrid algorithm, the DV component will use static metrics to recompute the failed paths to X. Notice that DV will invalidate all Scout paths that used link (P, R) but leave the valid paths untouched. In particular, the dotted curve from R to X remains valid while the curve from Y to P is invalid due to the link failure.

In this example, the path that the DV component calculates from Y to X extends a valid Scout path. The newly calculated path is denoted by the solid line from Y to S and

[‡]If X broadcasts Scouts before DV's link failure detection and that the Scouts calculate valid paths to X, then DV will not nullify the newly calculated Scout path because the path would appear to be a valid DV path.

RED HAT 1012
RED HAT V. CAS

the dotted line from S to X. Furthermore, because DV calculates static metric shortest path, the solid path from S to Y is the shortest, static metric path that connects Y to a valid Scout path to X.

3.1.4 Scout-LS Hybrid

This section describes the Scout-LS hybrid algorithm. LS, like DV, is a widely used and well understood routing algorithm. However, unlike DV, it computes paths in a centralized manner: a topology broadcast mechanism ensures that each router knows the current state of the entire network (i.e. topology and link cost), and paths are calculated by each router knowing the entire network topology. This centralized path calculation method posses slight problems to the hybrid algorithm when component failures occur. The basic hybrid algorithm is described below, and solutions to address component failures follow.

Scout-LS Algorithm

The Scout-LS hybrid, like the Scout-DV hybrid, requires very little modification to either algorithm. The LS component maintains its forwarding table, computes paths using the static link metrics, and recomputes routes on detection of link/node failures/recoveries in the usual manner to non-Scout generating destinations; LS processing of Scout paths is discussed below. For Scout generating destinations, Scout messages directly modify LS forwarding tables and compute dynamic metric paths to their initiating destinations. Like the Scout-DV, Scout is modified such that a destination's default designated neighbor is LS's next-hop neighbor to that destination. As proven earlier, this modification does not affect the validity of Scout paths.

The following subsections describes the LS integration with Scout. The main area of concern is the handling of Scout paths when the LS component detects network component failures. This problem and its solution are described below.

RED HAT 1012
RED HAT V. CAS

Component Failures

LS routers exchange topology information to compute shortest paths. Whenever a network failure occurs, the LS routers that detect the failure will broadcast this failure information, which then triggers a network wide path recomputation. Because Scout paths do not carry information as to the nodes/links they traverse, this means that whenever a Scout-LS router receives a topology broadcast indicating a node/link failure, it cannot determine whether a Scout path traverses the failed node/link. Thus if the LS component does not correct Scout paths invalidated by the failure, these paths will remain invalid in the number of Scout BI's needed to correct these paths. To speed the validation of Scout calculated paths, a mechanism is needed to correct invalid Scout paths. As in Scout-DV interactions, this Scout-LS interaction is an optimization because the Scout component will correct its paths in the following BI's.

Unlike failures, component recoveries do not pose similar problems because Scout paths affected by a recovery are still valid; therefore packets sent on these paths will reach their destinations. Thus, ensuring that Scout paths are aware of component recoveries is not as critical as in component failures. Because it is less critical and that relating Scout paths to LS paths is difficult, the Scout-LS's LS component does not alter Scout paths after node/link recoveries. Recall in Scout-DV, the DV component does "optimize" Scout paths after component recovery. This optimize was performed because the DV and Scout path computation mechanisms are very similar, allowing a natural implementation of this optimization.

For Scout-LS, the different possible approaches to cope with component failures are given below.

1. Do not update Scout paths. This means a Scout path could potentially be invalid for two Scout BI's. This is a reasonable solution in networks where the time to detect link/node failures are on the order of the Scout BI's.

2. Update all Scout paths to static metric LS shortest paths. This approach replaces

RED HAT 1012
RED HAT V. CAS

all Scout paths whenever LS recomputes paths due to a component failure. The disadvantage of this method is that it changes all Scout paths, even those that do not traverse the failed component(s).

3. Implement a DV style invalidation procedure. This approach implements a DV-like procedure to nullify Scout paths after component failures. In this method, when a failure is detected, Scout-LS routers that detect the failure will initiate the procedure to nullify all Scout paths invalidated by the failure. This DV-like procedure is proven to nullify all invalid Scout paths (Section 3.1.3). Notice that the procedure does not need to recompute the nullified Scout paths because the LS component already knows the static metric shortest paths to all destinations.

4. Attach source routes to Scouts. In this approach, Scouts maintain a source route of the path they advertise. Thus, whenever a component failure occurs, routers simply check if the link/node that failed is traversed by a Scout path; if so, the path is updated to the LS calculated static metric shortest path. However, the disadvantage of this approach is that it significantly increases the size of Scouts.

The four approaches listed above address link/node failures in different ways. The first approach simply allows the Scout mechanisms to recalculate alternate paths. The advantage of this approach is its simplicity and that it requires no changes to the Scout-LS algorithm. The principal disadvantage is that the paths to selected destinations may be invalid for 2 BI's. This is undesirable because traffic to selected destinations are assumed to be important, and therefore connectivity (i.e. valid paths) to these destinations should be established as soon as possible.

The second approach indiscriminately invalidates all Scout calculated paths. This approach is simple to implement and immediately establishes connectivity to the selected destinations. The drawback is that the invalidation causes a majority of Scout paths to unnecessarily revert to the static shortest path (assuming a majority of the Scout paths are not affected by the failure). This sudden change in routing behavior may have a negative

RED HAT 1012
RED HAT V. CAS

performance effect if the number of Scout paths are high [84].

The final two approaches accurately invalidate only the affected paths. The first approach uses the Scout-DV invalidation mechanism to delete Scout paths. This approach has the disadvantage that it requires significant additions to the Scout-LS algorithm. The fourth approach simplifies the path invalidation procedure but complicates the Scout component. This approach attaches a source route to every Scout. With a source route, determining whether a path traverses a link is trivial; however, the disadvantage is that Scout messages are no longer fixed size, significantly decreasing their efficiency.

The solution adopted in this thesis is to modify the fourth approach such that Scout paths are selectively and conservatively deleted while maintaining a fixed Scout size. The solution is as follows: a 64-bit field is added to hybrid-LS Scout (initially zeros) and each router is required to identify itself with a 64-bit ID. These router IDs do not have to be unique. Every time a router forwards a Scout, the Scout's 64-bit field is bit-ORed with the router ID. A router stores a Scout's 64-bit field if it uses the path advertised by the Scout. Whenever a link or node fails, the attached routers broadcast the failure along with their 64-bit ID. Upon receiving this LS broadcast, routers recompute their paths as usual, and for each Scout path, the path is invalidated if the broadcasting router's ID is contained in the 64-bit Scout field. This containment test is a simple bit-AND operation.

For example, consider a Scout passing through three routers with the following 8 bit IDs: `00110000`, `01001000`, `00010001`. The Scout's bit field will be `01111001`, the bit-OR of all the three routers. If a link attached to the second router fails, the router will broadcast an LSP that indicates the failure and that contains the router ID, `01001000`. Routers using the this Scout path will invalidate this path because the ID in this LSP is contained in the Scout's bit field. Using this method, all Scout paths that traverse a failed component is guaranteed to be detected. However, if a router with ID `01100000`, which is NOT on the Scout path, broadcasts a component failure, then this Scout path will also be invalidated. These false invalidations are called false positives.

Because this method is conservative, it may delete Scout paths that do not traverse the

RED HAT 1012
RED HAT V. CAS

failed component, as the previous example shows. Simulations show that if a router selects its ID by randomly marking two bits in the 64 bit field and if Scout paths are 15 hops long on average, then the number of false positives is around 15%. With the low false positive rate, this approach provides a good tradeoff between Scout efficiency and invalidation accuracy.

3.1.5 Algorithm Comparisons

One of the main differences between hybrid-Scout (i.e. Scout-DV or Scout-LS) and dynamic metric LS and DV algorithms is that the efficiency and effectiveness of Scout depends on the number of Scout generating destinations, Scout generation rate, and how much traffic those destinations receive. In contrast, with dynamic metric LS and DV, routing costs depend on the degree of congestion experienced in the network.

As stated earlier, in order for hybrid-Scout to effectively reroute congestion, the amount of traffic received by the Scout generating nodes must be "significant". In addition, in order for Scout to be efficient, the number of hot destinations must be "small". We experimentally quantify these two conditions in Section 3.2 and show that these conditions are typical of real networks.

In summary, the main features of the hybrid-Scout algorithms are:

1. Hybrid-Scout can selectively upgrade paths to a destination to use dynamic metrics. This is done by simply having the destination generate Scout messages.

2. Hybrid-Scout reduces route oscillation when compared to LS/DV with dynamic metrics. This is accomplished by computing dynamic metric paths to only selected destinations in a time staggered manner.

3. In hybrid-Scout, the desired route quality to a destination can be determined independently on a destination by destination basis. That is, hybrid-Scout can easily provide different path qualities to different destinations, depending on their importance and traffic usage. This adjustment is done solely on the part of the Scout generating node by changing its Scout BI.

RED HAT 1012
RED HAT V. CAS

4. Hybrid-Scout's routing traffic is independent of network traffic. Therefore, Scout routing traffic can be easily controlled by limiting the number of Scout generating nodes and their BI's. LS and DV routing costs must be bounded using trigger thresholds and hold-downs.

For this thesis, we assume that the set of "hot" destinations is statically known and configured to send Scouts. In the Internet, for instance, ISPs and popular Web sites are likely candidates. Should static configuration prove too inflexible, traffic monitoring can be used to maintain the set of hot destinations dynamically (see Chapter 11.2).

3.2 Simulation Results

This section presents extensive simulation results to evaluate the proposed hybrid Scout algorithms and to compare their behavior with dynamic metric DV and LS routing algorithms. The purpose of these experiments is to determine whether the hybrid-Scout algorithms deliver better overall network performance at lower costs. While network topology is a key parameter determining the performance of all four algorithms, a single class of topologies were chosen to study scaling effects: viz., Internet-like topologies. These topologies are characterized by a highly connected backbone network, to which multiple tree-like subnetworks are attached. The decision was made not only to simplify the experimental analysis but also to make the results relevant to present-day networks.

Since hybrid-Scout recalculates routes based on dynamic metrics only to a set of "hot" destinations (Scout generating nodes), a natural question that arises is how the routing cost varies as the number of hot destinations changes? And as the skew in the traffic distribution to these destinations is increased, is the improvement in route quality worth the extra routing cost? Answers to these questions are obtained by studying the performance of the considered algorithms under varying traffic distributions; in particular the number of hot destinations and the percentage of traffic directed to them. A related question is whether the improvement in route quality to hot destinations comes at the expense of those to non-Scout generating destinations. This section presents an experiment which addresses

RED HAT 1012
RED HAT V. CAS

this question by measuring non-hot destination network performance. For simplicity, traffic destined for Scout generating nodes (hot destinations) is called *foreground traffic* and traffic destined for other nodes *background traffic*.

The experimental analysis consists of studying the effects of (1) network size, (2) foreground traffic distribution, and (3) background traffic quantity on routing cost and overall route quality delivered by the different algorithms. To evaluate the algorithms according to these three parameters, different algorithms were tested on an Internet like topology to show the different dynamic behaviors of each algorithm and their relative performance and cost. Subsequently, background traffic is introduced to examine the behavior of these algorithms with varying traffic distributions.

3.2.1 Simulation Environment

The simulation environment is based on the "x-netsim" package from the University of Arizona [28], an execution-driven, packet-level network simulator. The simulator takes as input a description of the network topology, including link characteristics such as bandwidth and propagation delay, and a set of software modules that implement the various protocols running on the routers and hosts of the network. Simulation time advances according to the calculated transmission and propagation delay of packets in the network. Software processing in the routers and hosts are assumed to have zero cost.

The four routing protocols are implemented in the simulation testbed: the two hybrid Scout routing algorithms, a Distance Vector protocol similar to the Internet RIP protocol [74][§] and a Link State protocol similar to the Internet OSPF [41] routing protocol. In these experiments, only results for the hybrid DV-Scout algorithm are presented because the behavior of the two hybrid algorithms using dynamic metrics are exactly the same. The only difference is that the hybrid LS-Scout messages are 8 bytes larger than those used in hybrid DV-Scout.

[§]Our implementation includes the "split horizon" and "poison reverse" heuristics.

RED HAT 1012
RED HAT V. CAS

Dynamic metric LS and DV were implemented using trigger thresholds. As a router continually updates its out-going link's costs, it observes whether the difference between any of its link's current cost and the link's cost that was last advertised exceeds the threshold. If so, the router initiates a network wide shortest path recomputation. Trigger thresholds are expressed as *trigger percentages*. A trigger percentage of 50% means that a link's cost has to increase/decrease at least 50% of its dynamic range in order to trigger a routing update. The lower the trigger percentage, the more sensitive the algorithm is to link cost changes, the higher the path quality, but also the higher the routing cost (traffic). Refer to Sections 2.1 and 2.2 for more details on the dynamic metric DV and LS algorithm.

The LS and DV protocols are optimized such that route changes are propagated only in response to a triggered update. In the hybrid-Scout algorithm, the DV/LS component only calculates routes based on static metrics (i.e. it does not trigger on dynamic link cost changes) and only the Scout component computes paths using dynamic metrics. Because static metric paths are calculated once at the beginning of a simulation, the cost of the initial static route calculation are removed for all algorithms. Thus, routing costs presented are the direct result of changes in dynamic metrics.

The performance of the different routing algorithms is measured by average packet delay. Packet delay is the elapsed time between the sending of the packet and its reception. Each experiment is constructed such that packet drops do not occur; therefore, the average packet delay reflects the amount of queuing that packets experience.

Not only is packet delay a good end-to-end measure of path quality, it also reflects the precision in which a routing algorithm tracks network link costs. Given that link costs are based on queue lengths (described below) and queue lengths directly affect packet delay, an algorithm that minimizes a packet's delay minimizes the cost of the path that the packet traveled. Thus, the degree in which an algorithm tracks the network link cost changes is reflected in the observed average packet delay.

The queue size for each router's outgoing link is 100 packets long. A router samples the length of its queues every 10ms. A link's current average queue length is calculated

RED HAT 1012
RED HAT V. CAS

from the average of its instantaneous queue length with the previous running average. The dynamic metric cost function is linear from the static cost (average queue length of 0%) to 3 times the static cost (average queue length of 100%). All links have the same static link cost, and all three algorithms use the same static link cost, dynamic link-cost calculation mechanism, and link-cost function.

3.2.2 A Forest Topology

The purpose of the first experiment is to show the basic behavior and scalability of each algorithm. To make the experiment more relevant to today's networks, the DV, LS, and hybrid Scout-DV algorithms were simulated on the topology shown in Figure 3.5. The picture on the left shows a backbone topology consisting of 6 highly connected nodes. Three of the backbone routers are attached to a tree-like topology, shown on the right. The connections are denoted by a dotted triangle attached to a backbone router. The backbone links have three times the capacity of tree links and all links have equal static costs. There are also cross links between an interior router in each tree and the corresponding router in the neighboring tree, which is not shown in the figure. Four hosts are attached to each tree router and one host is attached to each backbone router. The network has a total of 111 hosts. For the purpose of the routing protocols, each host represents a separate destination. Leaf nodes transmit a stream of 10,000 packets to a host attached to a backbone router at 75% link capacity. Backbone routers with attached hosts that receive packets generate Scouts (i.e. they are initiate dynamic metric path computation to themselves).

Routing Behavior

Before examining the simulation results for the complete topology, we use one "tree" of the forest topology to demonstrate the routing behavior of each algorithm. For illustrative purposes, assume that $R1$ is the hot destination.

In this example, as the leaf nodes send their packets to $R1$, the queues for links $(R2, R1)$ and $(R4, R1)$ will begin to fill up. As they accumulate, the costs for those links increase.

RED HAT 1012
RED HAT V. CAS

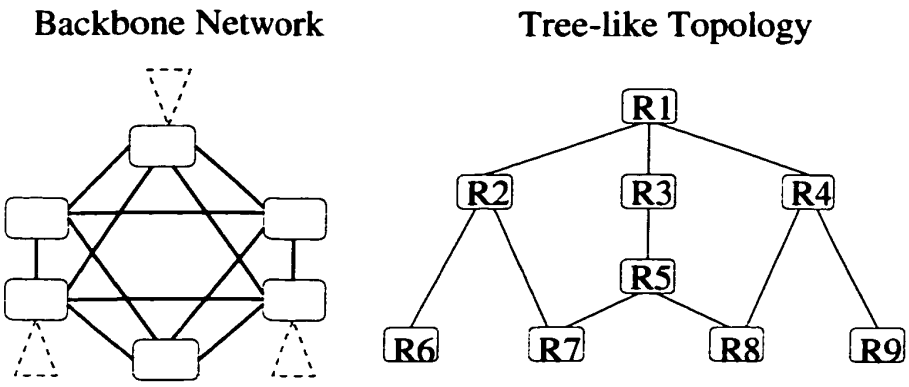


Figure 3.5 : The forest topology used in our experiments. The left-hand figure shows the backbone network, and the right-hand figure shows a tree network. The three backbone routers with a dotted triangle indicates that they have a tree network attached. Boxes represent routers and edges represent links between routers.

Once the cost of those links exceeds twice the link's base value, the shortest path from $R7$ and $R8$ to $R1$ shifts to the alternate path via $R3$, while $R6$ and $R9$'s packets still travel their primary path. After a while, the queue on $R5$ will also start to accumulate, and the traffic will shift back to the primary path. This shifting process continues until the transfers end.

The speed at which this path recalculation occurs depends on the sensitivity and granularity at which each algorithm is calculating the shortest path to $R1$. For LS and DV, this is parameterized by their trigger percentages: the lower the percentage, the faster the algorithm will trigger update messages upon link cost changes, and the faster it responds to congestion. For hybrid-Scout, it is controlled by its broadcast interval BI: the lower the BI, the better the calculated paths reflect the current link costs.

Routing Performance Versus Cost

Simulation results for the topology shown in Figure 3.5 is presented here. In this experiment, the number of hot destinations is varied by having leaf nodes transfer data to varying numbers of hosts attached to backbone routers. The traffic is uniformly distributed among these "hot" destinations. There are potentially two levels where routing based on dynamic

RED HAT 1012
RED HAT V. CAS

metric can increase performance. The first is within each "tree" (as illustrated above), the second is in the backbone network, where multiple alternate paths exist.

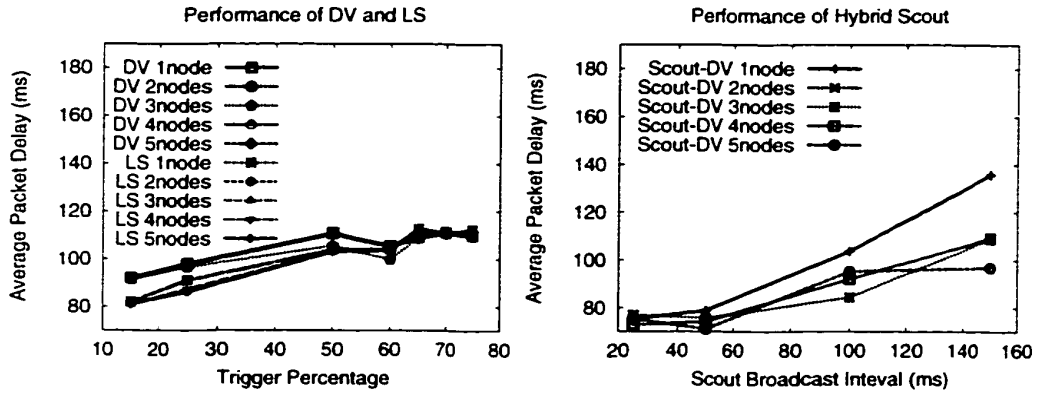


Figure 3.6 : The performance graphs for dynamic metric DV, LS and hybrid Scout algorithms. In the DV/LS graph, the top cluster of curves represent DV performance and the lower LS performance.

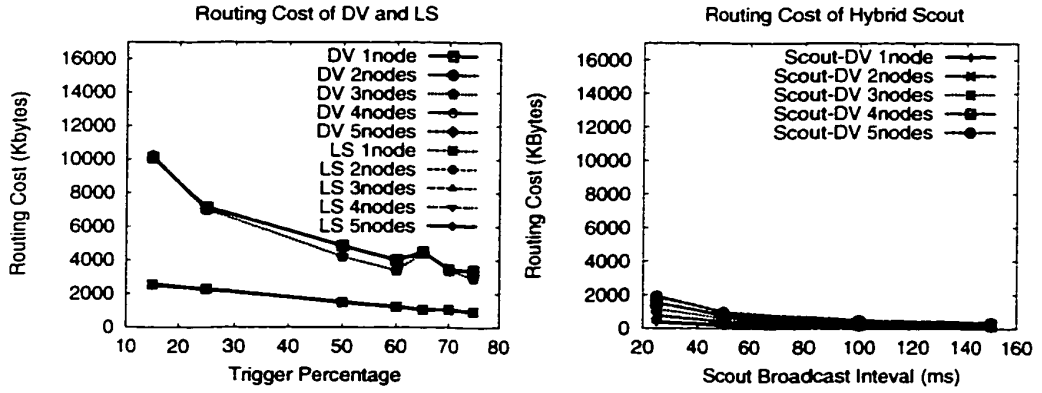


Figure 3.7 : The cost graphs for dynamic metric DV, LS and hybrid Scout algorithms. In the DV/LS graph, the top cluster of curves represent DV costs and the lower LS costs.

The performance and cost results are shown in Figures 3.6 and 3.7. In the performance graphs in Figure 3.6, the left graph shows average packet delay of DV and LS with dynamic metrics as a function of the trigger percentage, and the right graph shows hybrid-Scout's

RED HAT 1012
RED HAT V. CAS

performance as a function of the broadcast interval. The routing costs incurred by each algorithm are given in Figure 3.7. The label x-node denotes the number of distinct destinations to which the leaf nodes are sending packets. The amount of traffic injected into the network for the various number of destinations is fixed; only the distribution of the traffic differs.

The two figures show the effectiveness and efficiency of the hybrid-Scout algorithm on this topology. The performance and cost graphs show that hybrid-Scout achieves performance comparable to LS and DV at much lower routing cost. For example, the performance of LS/DV at 50% trigger percentage is comparable to hybrid-Scout's performance at a BI of 100ms and one hot destination. However, the routing cost of hybrid-Scout compared to LS and DV is approximately 15 and 40 times less, respectively. At five host destinations, hybrid-Scout, DV and LS achieve comparable performance at 100ms, 25%, and 50% trigger percentage; in this scenario, hybrid-Scout uses approximately 3 and 15 times less routing resources than LS and DV, respectively.

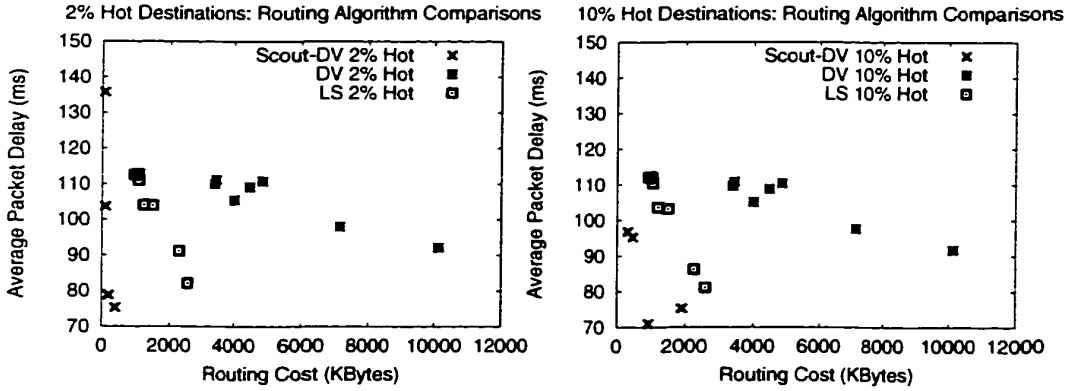


Figure 3.8 : Comparative performance of hybrid Scout, LS and DV for 1 and 5 hot destinations. The curves in these graphs are taken from graphs in Figures 3.6 and 3.7.

To highlight this cost-performance tradeoff, Figure 3.8 shows the routing performance versus routing cost of each algorithm with 1 and with 5 hot destinations. The x-axis denotes the routing cost, and y-axis the performance at that cost. Results are better the closer they

RED HAT 1012
RED HAT V. CAS

are to the origin, indicating good performance at low cost. Compared to LS and DV, hybrid-Scout consistently achieves better performance at lower costs.

As seen from the left graph showing one hot destination, hybrid Scout significantly outperforms both DV and LS on this topology. The graph clearly shows that for the same performance, hybrid Scout uses significantly less network resources than both LS and DV. For example, at a packet delay of 105000us, hybrid Scout requires approximately 100Kbytes of routing traffic while LS and DV require approximately 1224Kbytes and 4004Kbytes, respectively. The right graph confirms that hybrid Scout also outperforms LS and DV with 5 hot destinations.

Routing Cost Versus Network Size

The routing cost for hybrid-Scout and LS algorithms are both $O(L)$, where L is the number of links. The size of each Scout packet is constant (8 bytes for DV-Scout, 16 for LS-Scout) whereas the size of LS packets is proportional to the number of outgoing links per router. In addition, one hybrid-Scout broadcast ($O(L)$) recalculates all the current least-cost paths to a Scout originating node. In LS, however, if there are several points of congestion (say m points), every router at those points of congestion needs to perform a routing flood to recalculate the least-cost path ($O(mL)$).

DV is less scalable than either hybrid-Scout or LS on this type of topology because the size of each routing update is proportional to the number of destinations. The more destinations a network has, the higher the routing cost. With the network used in the experiment, the size of each DV packet is approximately 600bytes, compared to 32bytes for a LS packet.

In addition, since LS and DV compute shortest paths to all destinations simultaneously, they are more likely to cause massive route shifts (many paths simultaneously rerouted to a common set of links). This shifting of many routes may result in congestion, causing additional triggered updates. In hybrid-Scout, the shifting is less severe, because 1) hybrid-Scout does not recalculate dynamic metric paths to all destinations and 2) paths calculations

RED HAT 1012
RED HAT V. CAS

for different destinations are independent and staggered in time.

Table 3.1 experimentally shows each algorithm's scalability in routing cost with network size. For LS and DV, the trigger percentage was 15% and for hybrid Scout, there was 1 hot destination with BI of 25ms. The tree topology is the same as the one in Figure 3.5 and the different numbers of trees are configured as shown in Figure 3.5.

Tree Size	Hybrid-DV-Scout	Distance Vector	Link State	# of hosts	# of links
1-tree	52 Kbytes	35 Kbytes	60 Kbytes	9	10
2-trees	175 Kbytes	1,785 Kbytes	745 Kbytes	68	22
3-trees	385 Kbytes	10,120 Kbytes	2,550 Kbytes	111	45

Table 3.1 : The scalability characteristics of hybrid-Scout, Distance Vector, and Link State routing algorithms.

As the table shows, the scaling characteristics of routing cost for hybrid Scout and LS are relatively linear with the network size. LS is higher because the number of congestive points also increases with the increase in network size. The number of hosts attached to each router increased from 1 to 4 between the 1-tree and 2-trees topology. Thus the routing cost for DV increased significantly between the two topologies. This table confirms the above cost analysis.

Sensitivity to Traffic Distribution

For the hybrid Scout algorithm, the number of Scout generating destinations and the amount of traffic destined to them have a large impact on the algorithm's performance. To examine hybrid Scout's sensitivity to these two parameters, this experiment varies the number of hot destinations from 1 to 5 (approx. 1% to 5% of hosts) and the percentage of traffic destined for these destinations.

With respect to performance, Figure 3.6 shows that hybrid-Scout actually performs better when there are multiple "hot" destinations. This is because Scout calculates least-cost paths to different hot destinations at uncorrelated times, which allows hybrid-Scout to

RED HAT 1012
RED HAT V. CAS

split traffic: paths to destinations that would have shared the same set of links if calculated simultaneously may not, because later route calculations take into account the effect of earlier route changes on network load.

The amount of traffic splitting that hybrid-Scout can achieve depends on the difference in path calculation times. If paths to two neighboring destinations are calculated simultaneously (and thus based on the same link costs), the likelihood that their paths will have common links is high. On the other hand, if their paths are calculated at different times and some link costs change in the mean time due to previous route changes, then their paths will have less links in common. This fact explains why hybrid-Scout's traffic splitting is more prominent at higher BI's. At higher BI's, the mean time between calculations to different nodes are higher; therefore link costs are more likely to reflect the effects of previous route changes. As a result, congestion causing traffic is more likely to be split and network performance is improved.

Hybrid-Scout's routing cost is proportional to the number of Scout generating destinations. As seen in Figure 3.7, the routing cost for five Scout generating nodes (approx. 5% of the destinations) is exactly five times the cost with one Scout generating node.

The performance and cost of LS and DV are largely insensitive to the number of hot destinations[¶], since these algorithms calculate shortest paths to all destinations. The cost of updating paths to one hot destination is the same as updating paths to all destinations. This property manifests itself as coinciding lines in the LS/DV cost graph in Figure 3.6. LS and DV are able to split traffic to a lesser degree because they compute all pairs shortest paths simultaneously, as indicated by the relatively close performance curves in Figure 3.6.

To see the cost-performance ratios of each algorithm, the right graph in Figure 3.8 shows the relative ratios for 5 hot destinations. Again, one sees that hybrid Scout is able to provide better performance at lower costs than LS and DV, and that LS is more efficient than DV.

[¶] The performance and cost vary slightly because the traffic and congestion pattern change with the number of destinations.

RED HAT 1012
RED HAT V. CAS

3.2.3 Background Traffic

The purpose of this next experiment is to test 1) whether the benefits of hybrid-Scout observed in the previous section are maintained in the presence of background traffic, and 2) whether the performance of background traffic suffers with the hybrid-Scout algorithm.

The same network topology is used as in the previous section. In this experiment, the number of hot destinations is fixed at three (i.e. roughly 3% of the destinations are sending Scouts), and the foreground traffic is fixed at 50% tree-link capacity. At this rate, the foreground traffic by itself does not cause any network congestion; any congestion and subsequent rerouting in the network is caused by the additional background traffic.

To add background traffic, each host in the network repeatedly chooses a node at random and sends 100 packets to that node at 25% tree-link capacity. The amount of background traffic is controlled through the frequency of these transmissions. Increasing the background traffic increases the total amount of traffic injected into the network. Background traffic percentage is defined as the ratio between the amount of background traffic versus the total network traffic.

The experiment uses a trigger percentage of 15% for DV and LS and a BI of 25ms for hybrid-Scout. These parameters were chosen based on their comparable performance. The average packet delay for the foreground and background traffic are shown in Figure 3.9.

The left graph in Figure 3.9 shows the packet delay experienced by foreground traffic. As the overall network traffic increases (marked by the increase in background traffic), packet delay also increases. The results also show that hybrid-Scout's ability to reroute and split traffic allows it to achieve lower packet delay than LS and DV. We conclude that the performance benefits achieved by hybrid-Scout are maintained in the presence of background traffic.

The right graph shows packet delay experienced by the background traffic in the same simulations. The packet delay for the background traffic is less than the foreground traffic delay because most of the traffic do not encounter congested links. The fact that the background traffic performance of hybrid-Scout is comparable to DV and LS shows that the

RED HAT 1012
RED HAT V. CAS

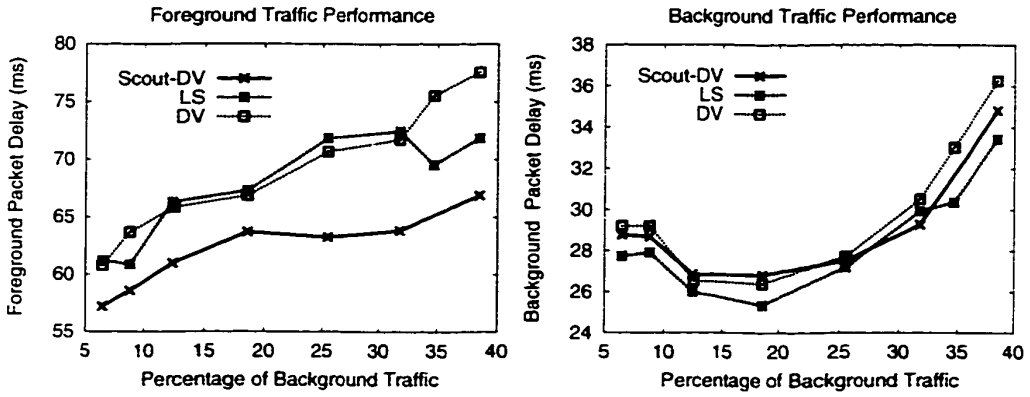


Figure 3.9 : The foreground and background performance in the large network. The percentage of background traffic is the percentage of traffic in the network that is destined for non-hot destinations.

increase in route performance for selected destinations obtained by hybrid-Scout does not come at the expense of non-selected destinations. The intuition is that by shifting traffic that contributes most to congestion (foreground traffic), the remaining traffic on those congested links are not likely to continue causing congestion. Hence, the network performance increases for non-selected destinations as well. The benefits obtained by hybrid-Scout for foreground traffic also indirectly benefit background traffic, as long as there is "enough" foreground traffic. The question of what is enough traffic is addressed in the next experiment.

An interesting feature in the background traffic delay graph is the dip near 12% background traffic. At less than 12%, the background traffic that traverse congested points is not enough to cause much rerouting; thus, packets experience queuing. Above 12%, enough congestion accumulates to cause rerouting, hence the packet delay is decreased because congestion is relieved. Of course, the packet delay again increases with background traffic, because the benefits of rerouting are offset by the increase in traffic. The dip at 12% is not observed in the foreground graph because the congestion that causes rerouting degrades the foreground performance such that there is a net increase in delay as a result of rerouting. The background traffic, on the other hand, always encounters congestion; therefore it can

RED HAT 1012
RED HAT V. CAS

only stand to benefit from rerouting.

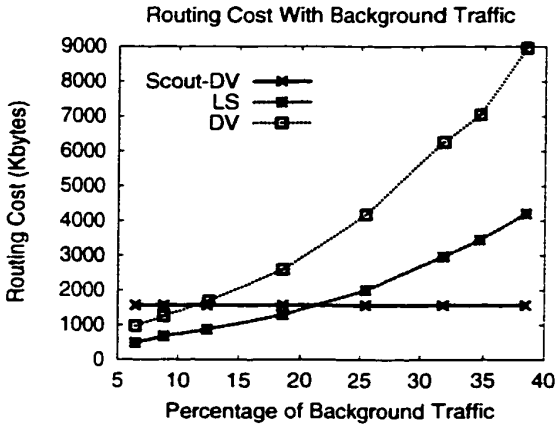


Figure 3.10 : Routing costs for the experiment in Figure 3.9.

Figure 3.10 shows the cost of each routing algorithm in this experiment. The hybrid-Scout algorithm exhibits a constant routing cost with different levels of network traffic. LS and DV's costs increase with the amount of traffic, with DV increasing most rapidly.

The cost graph shows an important characteristic of the LS and DV triggering mechanism: as the network utilization increases (marked by the increase in background traffic percentage), the LS and DV triggered updates also increase. This is undesirable because these updates compete with data packets for link bandwidth and router CPU at a time when links are already heavily utilized and routers are busy forwarding packets. This additional network load increases the probability of severe congestion and packet loss.

The cost graph also shows that LS and DV use less routing resources than hybrid-Scout at low network utilization. Here, only a few links are getting congested, resulting in few triggered updates. However, the advantages of this behavior under low network utilization are not as important as the disadvantages under high utilization levels. Since most links and routers are underutilized at low network utilization, the presence of additional routing traffic does not have a significant adverse impact on network performance.

RED HAT 1012
RED HAT V. CAS

3.2.4 Foreground Traffic

One of the premises of the hybrid-Scout algorithm is that the amount of traffic received by hot destinations must be "significant enough" for hybrid-Scout to be able to effectively reroute congestion, and that the number of selected hot destination must be "low enough" for hybrid-Scout to be efficient. The following experiment quantifies these conditions.

In this simulation, the same experimental setup as in the previous experiment (3 hot destinations) are used. However, the total amount of traffic in the network is kept constant while varying the distribution of foreground and background traffic. The performance and cost are shown in Figure 3.11. Foreground traffic percentage is the ratio of foreground traffic to total network traffic.

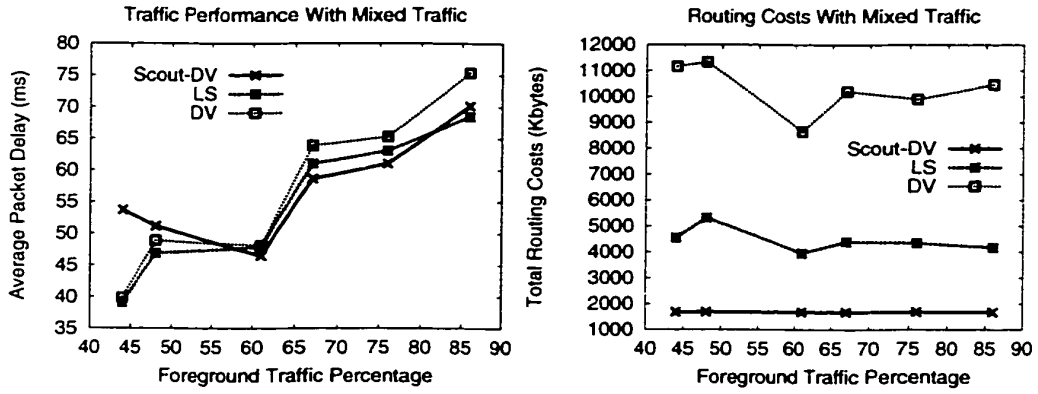


Figure 3.11 : Network performance and routing cost. The amount of traffic injected is held constant while foreground and background traffic percentages vary.

The performance graph (left graph) in Figure 3.11 shows the average packet delay of both the foreground and background traffic. As the foreground traffic percentage increases, more traffic is directed to fewer destinations, causing more packet queuing and higher average packet delay for all destinations. The performance graph also shows that hybrid-Scout is able to achieve performance comparable to LS and DV whenever the foreground traffic accounts for greater than 50% of the network traffic. Notice that hybrid-Scout's performance is significantly worse than LS and DV at 45% foreground traffic. This is because

RED HAT 1012
RED HAT V. CAS

Hybrid-Scout can only reroute paths to hot destinations; therefore it cannot eliminate congestion when these destinations only account for a minority of the traffic.

The cost graph (right graph) in this experiment shows that in the simulated network, where 3% of the hot destinations are transmitting Scouts, the routing costs of hybrid-Scout are around 3 to 4 times less than LS and an order of magnitude less than DV.

The routing costs for LS and DV actually increase at 45% and 50% foreground traffic. The reason is that the background traffic is causing minor congestion at many points in the network (as opposed to mainly on paths to the hot destinations when foreground traffic dominates), thus triggering more LS and DV updates.

With the results of this experiment, the two questions posed earlier in this section can be answered:

1. What fraction of total traffic do hot destinations have to receive in order for hybrid-Scout to adequately reroute congestion?

   From our experiments, the answer is at least 50%. The intuition is that if the dynamic routing algorithm is able to control 50% of the traffic, the likelihood that the remaining 50% continues to cause serious congestion is low.

2. How many destinations can generate Scouts for the hybrid-Scout algorithm to be efficient?

   In our topology, if 10% of the nodes generate Scouts (in a highly utilized network), the cost of hybrid-Scout is comparable to LS. That is, hybrid-Scout is cost effective as long as the number of "hot" destinations is below 10% of the total number of nodes in the network. Note that hybrid-Scout's efficiency also depends on the network utilization. As shown in the previous experiment, at low utilization LS/DV tend to be more efficient, and hybrid-Scout is better at higher utilization levels. However, efficiency at low utilization is not as critical as the efficiency at high utilization levels.

Recall that in our study of Internet traffic locality, 1% of the destinations account for over 50% of the network traffic. This indicates that in the Internet, having the hottest 1% of

RED HAT 1012
RED HAT V. CAS

the destinations generate Scouts will be as effective as LS or DV with dynamic metrics in increasing network performance. However, the cost of hybrid-Scout under these conditions should be approximately an order of magnitude less than LS and 2 orders of magnitude less than DV.

3.2.5 Path Approximations

As stated in Chapter 2, dynamic metric routing algorithms may not converge in scenarios where rate of link metric change is faster than the algorithm's convergence time. In these scenarios, an algorithm is correct if it tracks the network state and computes the intended paths over the algorithm's current network view of the network. As shown in Section 3.1.2 and Chapter 2, dynamic metric DV, LS, and Scout algorithms are correct in this sense.

The Scout algorithm (and thus the hybrid-Scout algorithm) converges to least-cost paths in at most l broadcast intervals after link metric changes, where l is the number of hops on the longest calculated path. This is a loose upper bound: in our simulations, the Scout always converges in at most three BI's. Thus, in practice, if the frequency of link cost changes is less than 2–3 BI's, then the dynamic metric Scout algorithm will converge to the least-cost paths. Unfortunately, link cost changes in most dynamic metric networks are much more frequent than Scout BI's.

In comparison to dynamic metric LS and DV, the converge of hybrid-Scout may take longer for sufficiently large BI's. However, it is important to note that unlike Scout, LS and DV must use trigger thresholds and hold-downs to limit the frequency of metric changes and maintain stability. Therefore, even if LS and DV technically converge while Scout does not in a given scenario, the calculated paths are, in both cases, approximations of the actual shortest paths because the cost metrics used in LS/DV calculations are not the actual instantaneous link costs. In addition, note that the metrics used in our simulations to determine routing cost and performance take into account effects of route accuracy and stability.

To see hold-down's effect on route quality, Figure 3.12 compares the cost and per-

RED HAT 1012
RED HAT V. CAS

formance of LS and DV using hold-downs. This experiment uses the same experimental configuration that produced Figures 3.9 and 3.10. Here, the hold-downs for DV and LS are set to 10-12 and 4-5 triggers per second respectively, and the hybrid Scout parameters are unchanged.

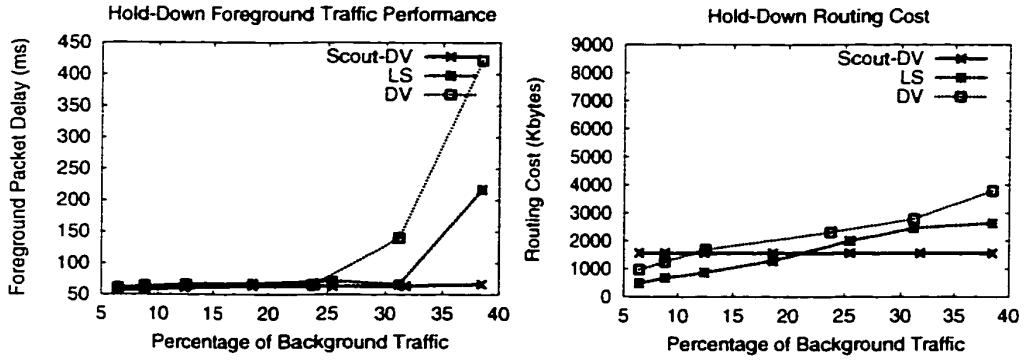


Figure 3.12 : Routing performance and cost using hold-downs. The graphs represent the performance and cost of LS/DV algorithms with hold-downs compared to hybrid-Scout. The experiments used the same configuration as ones used in Figures 3.9 and 3.10.

The right-hand graph in Figure 3.12 shows the cost incurred by each algorithm. Notice that, compared to Figure 3.10, the routing cost of LS and DV are much lower at high network utilization. This is because the hold-downs prevent excessive routing triggers. However, the reduction in routing cost of using hold-downs comes at the expense of routing performance. The left-hand graph shows that the performance of both LS and DV is dramatically worse using hold-downs. This confirms the argument that by delaying path computation, hold-downs force LS and DV to more coarsely approximate dynamic metric least-cost paths (i.e. more coarsely track the network state), thereby degrading network performance.

3.3 Hybrid-Algorithm Summary

Dynamic metric routing has been shown to increase network performance in real networks. However, they are currently not in wide use due to the dangers of routing instability and

RED HAT 1012
RED HAT V. CAS

high routing costs. This chapter presents a new approach to routing using dynamic metrics that promises to overcomes the above limitations. The approach is based on the observation that real network traffic exhibit a high degree of destination locality. Analysis of Internet traffic traces shows that 1% of the hot destinations receive over 50% of the network traffic.

The proposed algorithm, hybrid-Scout, is able to calculate paths based on dynamic link metrics to selected destinations, while paths to other destinations are calculated by traditional routing algorithms using static link costs. Extensive simulations were performed to determine the effectiveness and efficiency of hybrid-Scout in rerouting congestion compared to dynamic metric LS and DV algorithms, and under what conditions. Through simulations, these questions are answered using an Internet-like topology consisting of over 100 host networks and 30 routers. In summary, our simulations show that:

1. Hybrid-Scout is effective at rerouting congestion if at least 50% of the network traffic is destined to hot destinations (i.e. hybrid-Scout generating destinations). Hybrid-Scout is more efficient than both dynamic metric LS and DV if no more than 10% of the network nodes are generating Scouts.

2. Hybrid-Scout is more scalable than LS and DV with dynamic metrics. While achieving comparable network performance (measured in packet delay), hybrid-Scout has substantially less routing cost (in routing message bytes), from 4-5 times to 1-2 orders of magnitude.

3. The selective update mechanisms of hybrid-Scout better splits congestion-causing traffic, which reduces route oscillations. This splitting is achieved by 1) only rerouting selected destinations based on dynamic link metrics and 2) by calculating new routes for those selected destinations in a time staggered manner.

4. Hybrid-Scout's routing costs are stable even under high network utilization levels. This ensures that hybrid-Scout does not exacerbate network load during high network utilization. This is in contrast to dynamic metric LS and DV algorithms that tend to increase routing traffic at high network utilization levels.

RED HAT 1012
RED HAT V. CAS

Chapter 4

Multipath Routing

The second main contribution of this thesis is the development of a complete static metric multipath routing model: from calculating multiple paths between nodes to end-host methods that utilize multiple paths to increase performance. Like dynamic metric single path routing, multipath routing offers potential performance increase over single path routing by better utilizing network resources. This introductory chapter presents the multipath routing model* and describes the various components needed for its implementation.

The chapter begins with multipath routing definitions to facilitate later discussions. Section 4.2 uses these definitions to discuss the advantages and disadvantages of multipath routing. Finally, Section 4.3 summarizes the necessary multipath routing components for increased network performance. These components are individually addressed in following chapters.

4.1 Multipath Routing Definitions

A glossary of terms to describe multipath routing models is presented in this section. The terms are structured to reflect the key parameters that define multipath networks and influence their performance. These parameters include

1. Basic definitions. The basic terms that describe a multipath routing model.

2. Path specification and calculation. *Path specification* describes the properties of the paths to be calculated between nodes. Example specifications are to calculate node

*For convenience, the term "multipath routing" is used to denote static metric multipath routing, unless otherwise specified.

RED HAT 1012
RED HAT V. CAS

disjoint paths, shortest K paths, and maximum flow paths between nodes. *Path calculation* is the actual algorithm that calculates the specified paths.

3. Multipath types. This parameter describes the paths a multipath routing algorithm provides between nodes. The two multipath types are *multi-service* paths where the routing algorithm provides different paths with different characteristics (example characteristics are high throughput and low delay) and *multi-option* paths where an algorithm provides multiple paths with the same characteristic. A routing algorithm that provides either or both multipath types is considered multipath routing algorithm.

4. Usage layer. The software layer responsible for using multiple paths to a given destination. This layer manages multiple paths by dictating which data packet should be sent on which path and when. Example layers in today's Internet protocol stack are the network, transport, and application layers.

5. Multipath usage. The way an end-host (or the usage layer of the end-host) uses multiple paths to transmit data.

Basic Definitions

The basic definitions of multipath routing are given here. First, a *multipath routing model* is defined as a routing model where the routing algorithms provide potentially multiple paths between node pairs and allows the end-hosts (or applications) to choose how to use these paths. We require that end-hosts have control over which path to use because this control offers the most flexibility in using multiple paths. This flexibility allows applications to use multiple paths in ways to best maximize their performance.

Using this definition, dynamic metric, single path routing algorithms do not implement the multipath routing model; although these algorithms may route packets between a node pair on different paths, end-hosts do not control the path a particular packet will travel. For

RED HAT 1012
RED HAT V. CAS

the same reason, networks with backup paths, such as telephone networks [150], do not implement the multipath routing model.

A *multipath routing algorithm* refers to a routing algorithm that provides multiple paths between nodes so that data sent on a path travels that path through the network. A *path set* refers to the set of paths that a routing algorithm calculates for a particular network topology, and *multipath networks* are networks with routers that execute a multipath routing algorithm. That is, multipath networks are networks that offer multiple paths between node pairs.

Path Specification and Calculation

In order to calculate paths between nodes, one must first specify the characteristics of the paths to calculate. The different path characteristics depend on the intended use of the paths. For example, paths intended to maximize end-to-end throughput should be specified such that, for any node pair, the aggregate throughput obtain on multiple paths is maximized. In contrast, paths intended to minimize transmission delay should be specified such that, at any given time, there exist at least one low delay path between node pairs. *Path specification* specifies the characteristics a particular path set.

A *path calculation algorithm* is the algorithm that actually calculates the paths specified by path specification. A practical path calculation algorithm takes into account the operating resources and environmental constraints such as the distributed nature of a network. Examples of path calculation algorithms are Dijkstra's shortest path algorithm [54], Topkis's initial link disjoint paths algorithm [156], and the Bellman-Ford distributed shortest path algorithm [22, 35].

Multipath Types

Path type specifies the relationship among the paths a routing algorithm provides between node pairs. There are two path types: *multi-service* and *multi-option*. The first path type, multi-service paths, denotes paths between nodes that have different characteristics (i.e.

RED HAT 1012
RED HAT V. CAS

different path specifications). Example services that a network could provide are low delay and high bandwidth path services. Since applications may have different demands on the network, providing paths with different characteristics allows applications to choose paths that best fit their communication demands.

The second path type, multi-option paths, denotes the scenario where a routing algorithm provides multiple paths with the same path service. For example, an algorithm might provide four multi-option paths for the high bandwidth path service. That is, each end-host has, to each destination, four paths that can provide high bandwidth to a destination.

Networks that support multi-service and/or multi-option paths are called multipath networks. For example, a multipath network can be one that provides multiple service paths with only one path in each service (multi-service, single option), or one that provides only one path service with many paths within that service (single service, multi-option).

This thesis considers the general multipath routing model where networks provide multiple services each with multiple paths.

Usage Layer

Usage layer refers to the highest protocol layer responsible for managing multiple paths. The levels applicable in today's Internet protocol stack are the network, transport, and user/application layers. If the usage layer is the network layer, then it is the responsibility of this layer to decide which path a packet should travel and to do this transparently to the protocol layers above. Similarly, if the multipath usage layer is the transport layer, this means the transport layer has the freedom to send data on multiple paths.

The protocol layer that manages multiple paths needs to effectively use these paths to increase performance. The choice of usage layer depends on the tradeoffs between flexibility, performance, and the software engineering issues of implementing multipath management at a particular protocol layer. Chapter 7 discusses usage layers in more detail.

RED HAT 1012
RED HAT V. CAS

Multipath Usage

Usage mode characterizes how multiple paths are used. There are two prototypical multipath usage modes: using paths concurrently or one at a time. The choice of which usage mode is application specific. For example, for applications interested in maximizing throughput, such as FTP, the right usage mode is to use all paths concurrently to obtain the aggregate bandwidth of all available paths. On the other hand, the appropriate usage mode for delay-sensitive applications, such as Telnet, is to use one path at a time, preferably the path with the lowest delay. Another application is one that needs to send urgent messages; here, the appropriate mode may be to send urgent messages on multiple paths concurrently, minimizing the message delivery time to the minimum time of all the paths used.

In general, usage mode varies with application needs. In the foreseeable future, applications might require paths from different path services at the same time, thereby requiring different usage modes. For example, a Web session may have concurrent large file transfers and time-critical user interactions. In this scenario, the appropriate usage mode may be to use paths from different services and multiplex data among these paths according to the type of the data.

One expects that as network applications become more sophisticated, their usage modes will increase in complexity as well. The multipath routing model is able to accommodate these complex applications because it does not place restrictions on usage modes. Thus, nodes are allowed to use the offered paths in ways that best fit their communication needs.

4.2 Multipath Routing Overview

This section presents a conceptual overview of the multipath routing model in order to describe the potential benefits and costs of providing multiple paths and to highlight the components needed to make multipath routing viable.

This section is organized as follows. The first two subsections present the advantages and disadvantages of the multipath routing model and argue that the flexibility of the model

RED HAT 1012
RED HAT V. CAS

offers significant network performance gains that outweigh the potential disadvantages. The latter two subsections, 4.2.3 and 4.2.4, present the key multipath implementation issues. Solutions to resolve these issues are addressed in succeeding chapters.

4.2.1 Multipath Advantages

The multipath routing model offers applications the ability to increase their network performance. Because of its multi-service paths, multi-option paths, and end-hosts' freedom to use these paths, the model provides a flexible interface to network resources that enables applications with varying network demands to increase their performance.

In general, multipath performance improvements are obtained in two ways. First, multi-service paths allow an application to use paths within a service that best suit the application's communication needs. Second, multi-option paths provide more network resources per path service, allowing applications to aggregate these path resources. These two general approaches are discussed below.

Providing the Right Paths

A multipath network with multi-service paths improves network performance because it allows applications to choose the paths that best suit their communication style. For example, an application such as FTP can improve its performance, measured in throughput, if it uses high-bandwidth service paths. Similarly, an interactive application, such as Telnet, can increase its performance, measured in response time, if it uses low-delay service paths. Because network performance depends on application demands, networks that provide paths with characteristics that fit these demands will be able to increase application network performance. Since network demands vary with applications, the generality of a multi-service paths allows a multipath network to satisfy the needs of different applications.

Providing the appropriate paths to increase performance will become more significant as the diversity of network applications increases. For example, in the foreseeable future, network applications such as IP telephony, real-time medical imaging, and video confer-

RED HAT 1012
RED HAT V. CAS

encing will become more prevalent. These applications need paths with very different characteristics from those of traditional applications. Specifically, many of these new applications need paths with QoS and real-time guarantees. In this environment, a multi-service network might provide paths with different delivery guarantees, allowing applications to select the paths that best suit their needs.

Notice that in a single path routing model, it is in general not possible to customize the one path between a node pair with characteristics suitable for all applications. In practice, single path routing algorithms calculate paths that compromise between throughput and delay [94]. Although the paths generated by this single path compromise are sufficient for today's applications, it seems unlikely that these paths can effectively support future applications that need paths with radically different characteristics.

Aggregating Multiple Paths

Multi-option paths increase application performance by giving applications the freedom to use multiple paths within the same path service. Performance improvement is obtained in two prototypical ways: 1) aggregating resources of multiple paths and 2) selectively using the best available path. The descriptions of these two methods are given below.

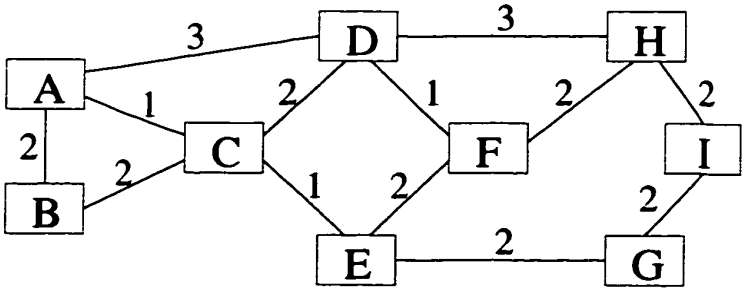


Figure 4.1 : An example network topology. Boxes represent network nodes and edges represent links. The number above each link shows the link's capacity, and all links have uniform delay.

Using multiple paths, the simplest method to increase performance is to aggregate path resources. For example, consider node B maximizing its throughput to node D in Fig-

RED HAT 1012
RED HAT V. CAS

ure 4.1. In this figure, boxes denote nodes and the number above each link denotes the link's capacity. In a single, shortest path network, the maximum bandwidth between B and D is 2 units: either the path (B, A, D) or (B, C, D), but not both. However, a multipath network can provide both paths to B, allowing B to send data to D at 4 units of bandwidth. Similarly, an QoS application can increase the probability that the network satisfy its QoS request by combining the QoS reservations it makes on multiple paths. In general, this style of resource aggregation can benefit any application that can use multiple paths in parallel.

End-hosts can also use multi-option paths to increase performance by selectively using the available paths. For example, consider an application interested in low delay. In a single path model, the application has no choice but to incur the delay of the one path provided by the network to its destination. On the other hand, an application in a multipath network can attain lower communication delays by probing the delays among the available paths to the destination and then choosing the minimum delay path. Moreover, if low latency is one of the multi-service paths, a node can choose among the paths in that service category.

In summary, end-to-end performance is measured with respect to application demands; thus, different applications maximize their performance differently. These differences are accommodated by multipath routing model's multi-service and multi-option paths. This model offers a flexible interface to network resources that allows different applications to increase end-to-end performance compared to single path routing models.

4.2.2 Multipath Disadvantages

Routing algorithms play a major role in end-host resource usage because they allocate network resources (in terms of paths) between nodes. By construction, multipath routing algorithms offer more network resources to end-hosts than do single path routing algorithms, both to specific destinations and to sets of destinations. The previous section showed that the advantage of multipath routing is that these additional resources can be used to increase end-host performance; however, a potential disadvantage is that these same resources can also be used by a greedy or malicious user to deny other users their fair share of network

RED HAT 1012
RED HAT V. CAS

resources. This section argues that although multipath routing may exacerbate resource denial, the problem is actually orthogonal to single or multipath routing.

For example, consider a FIFO datagram network that does not place any restrictions on how much data end-hosts can send. In this network, excessive resource consumption and denial of services cannot be prevented [52]. In today's Internet (a single path, FIFO datagram network), nodes can blast packets to random network destinations, consuming a significant amount of resources and drastically degrading overall network performance. Although malicious users in a FIFO multipath datagram network could deny more resources than in a FIFO single path data network, the fact remains that FIFO datagram networks do not have mechanisms to prevent resource abuses, whether single or multiple paths are provided.

In general, two approaches can prevent or reduce excessive resource denial: cooperative network communities and enforced network policies. In the cooperative approach, network users/applications agree not to consume excessive resources. The Internet uses this approach via congestion sensitive transport protocols (e.g. TCP [83]). These protocols attempt to share resources fairly by regulating their sending rates in response to the available bandwidth of the path they are using. Internet resource abuses are low because these protocols are used by most users [124]. The advantage of the cooperative community approach is that it does not require any network support – Internet routers do not need any additional mechanisms to prevent resource abuse because end-hosts voluntarily do not abuse resources. However, the disadvantage of this approach is that users (e.g. malicious ones) may not abide by the convention and thus can consume excessive resources. In order to control these users, a network needs mechanisms for admission and traffic control (described below). The important point is that the success or failure of the cooperative approach is independent of whether single or multiple paths are provided between nodes: if all hosts cooperate, excessive resource consumption will not occur in either single or multiple path networks. Similarly, if hosts do not cooperate, resource abuse can occur in both types of networks.

RED HAT 1012
RED HAT V. CAS

The second method to prevent/reduce resource abuse is for the network itself to enforce admission and traffic control policies [86, 135]. These networks enforce end-host resource usage by regulating the number of senders and/or the amount of data each sender sends. Regulating network traffic requires specific mechanisms in the network (e.g. in routers) to monitor and enforce end-host sending policies. For example, a pricing network implicitly discourages resource abuse by charging users for packets they transmit [121]. Users in these networks are unlikely to maliciously consume excessive resources because they have to pay for the resources they use.

Again, notice that the enforceability of traffic control in pricing networks is independent of whether single or multiple paths are provided between nodes. Users in a pricing network pay for packet transmission, regardless of whether they send their packets on different paths or on the same path. Other networks with mechanisms and policies to discourage/prevent excessive resource consumption can be found in [91,92,113,121,135]. Although the actual implementation of the admission and traffic control mechanisms may differ, the enforceability and effectiveness of admission and traffic control policies are orthogonal to the routing model.

The advantage of the network enforcement approach is that it does not rely on the cooperation of end-hosts; thus, it is much more robust and can prevent malicious users from abusing the network. The primary disadvantage is that it requires network mechanisms to regulate traffic and prevent abuse. These monitoring and enforcement mechanisms increase network cost and may decrease performance because of additional packet processing.

In summary, although multipath routing can exacerbate resource abuse in certain types of networks, the core issues of resource abuse are orthogonal to both single and multipath routing. That is, the fact that a network is prone to resource abuse is independent of whether it provides single or multiple paths between nodes. Because of this independence, the remainder of the thesis assumes that users are greedy but not malicious. That is, end-users greedily maximize their resource usage but attempt to avoid network congestion. This is the same end-user assumption used in today's Internet.

RED HAT 1012
RED HAT V. CAS

4.2.3 Multipath Implementation Cost

The advantages of multipath routing come at a cost. Recall that routing is a two-step process: 1) calculating paths and 2) forwarding data on those paths. Implementing these two routing tasks incurs the following three cost categories:

1. The cost of computing multiple paths
2. Per packet path specification overhead in bytes
3. Router overhead of processing and forwarding data packets.

The first category corresponds to the cost of path computation, and the latter two to the cost of forwarding data on the computed paths. The three costs are described below.

Computing Multiple Paths

The first cost category, computing multiple paths, is measured in terms of routing messages (in bytes) needed to propagate routing information and the router CPU time needed to compute multiple paths. The number of messages and the amount of CPU usage depends on the path calculation algorithm and the base routing algorithm.

For example, the number of routing messages needed to compute multiple paths heavily depends on whether the routing algorithm is based on LS or DV. In an LS environment, routing message overhead is generally low because path computations are done with the knowledge of the network topology. Thus, the number of messages needed to disseminate topology information is the same independent of whether multiple or single paths are calculated. In contrast, DV based algorithms use routing messages (in the form of Distance Vector packets) as the mechanism to propagate paths; therefore, computing multiple paths generally requires more DV messages than computing single paths.

The amount of router CPU time to compute multiple paths has similar dependencies. In LS, because path computations are centralized, standard complexity analysis of centralized algorithms suffice to measure LS router CPU usage. As examples, calculating multiple K

RED HAT 1012
RED HAT V. CAS

initial link disjoint paths takes $O(K * E * lg(E))$ and calculating the shortest K loop free paths takes $O(nE * lg(E))$ [58, 141, 156], where n is the number of nodes (or routers), and E is the number of network edges. In DV, the analysis is not so straightforward because path computations are distributed. In the worse case, the message and CPU complexities are exponential [25]. However, it has been shown that the average message and CPU complexities is $\Theta(nM^3(ln(M))^2)$ [23, 157], where M is the average number of neighboring routers. Efficient multipath calculation algorithms based on DV and LS are given in Chapters 5 and 6.

Specifying Multiple Paths

Because there are multiple paths between nodes, every packet needs to specify not only its destination, but also a particular path to that destination. This is in contrast to single path networks where a destination address uniquely specifies a packet's path. The second multipath cost category refers to this additional per packet cost of path specification. The specification cost is measured in the number of bytes needed in order to ensure that a packet travels its specified path. It is critical to minimize this cost because it is incurred on every data packet. Chapter 6 describes an efficient method for packets to specify a particular path to a destination.

Forwarding Multiple Paths

Finally, the per packet path specification implies that more router processing is needed to forward each packet. The additional processing is needed for a router to decide, given the packet's destination address and path specifier, which outgoing link the router should forward the packet to. This additional processing may slow router forwarding speed and decrease network performance; thus it is critical to minimize this processing time. The cost of this additional processing is called the router forwarding overhead. Not surprisingly, this overhead is closely tied to how paths are specified in data packets. The efficient path encoding method presented in Chapter 6 has low forwarding overhead.

RED HAT 1012
RED HAT V. CAS

4.2.4 Multipath Benefits

The previous subsections list the potential advantages, disadvantages, and costs of a multipath network; this subsection concludes the multipath discussion by addressing how the benefits of a multipath network can be obtained. To obtain multipath benefits, 1) multipath networks need to calculate appropriate paths, and 2) end-hosts need to effectively use these paths. These two properties are described below.

Path Calculation

The extent of performance improvement users can obtain from a multipath network depends on the quality of the calculated paths. For example, an application can obtain higher throughput *only if* the multiple paths calculated actually provide greater combined bandwidth than the one provided by a single path routing algorithm. Similarly, an application can increase its probability of establishing a QoS connection only if the calculated paths have, either individually or combined, a higher probability of satisfying QoS requests compared to the probability of a single path.

For example, consider the three paths (A, D, H, I), (A, C, D, H, I), and (A, C, D, F, H, I) from node A to node I in Figure 4.1. These three paths are not well chosen if A wishes to increase its network throughput to node I because all three paths traverse the same bottleneck link (H, I) with capacity 2. On the other hand, if a multipath network provides paths (A, D, H, I), (A, B, C, E, G, I), and (A, D, F, E, G, I), then node A has 4 capacity units to node I.

As the example shows, a multipath network must provide the "right" paths in order for nodes to obtain higher performance gains, where the right paths depend on the applications that use the paths. In general, providing quality paths is a two-step process. First, determine the type of path services to calculate (path specification), then develop efficient algorithms to calculate them (path calculation algorithm).

Two algorithms are developed in this thesis, one that calculates paths to maximize throughput and the other to minimize latency. The complexities of both algorithms are

RED HAT 1012
RED HAT V. CAS

linear in the number of paths calculated compared to computing single shortest paths. The algorithms are described in Chapter 5.

Path Usage

The second component necessary for end-hosts to obtain increased performance in multipath networks is effective end-host usage of multiple paths. The fact that a network provides quality paths between nodes does not necessarily imply that nodes are able to effectively use these paths to maximize performance. This subsection shows the importance of effective multipath usage and its impact on network performance.

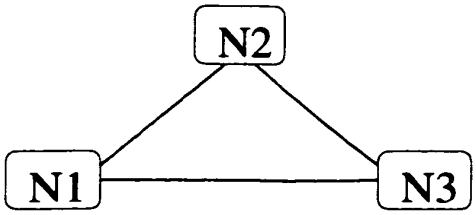


Figure 4.2 : A simple three node network with full-duplex links. All links have equal bandwidth and latency.

Consider the simple network in Figure 4.2 where all links have equal capacity and delay. Here, the multipath routing algorithm calculates two link disjoint paths (the one link and two link paths) between every node pair. Notice that the calculated paths provide the optimal paths for maximizing throughput. In this setting, the two paths calculated provide twice the capacity between nodes compared to single path routing; however, the effective throughput between nodes depends on how each node uses its paths.

To show the potential harm of naively using multiple paths, this three node network was simulated using TCP and a non-congestion aware, multipath striping protocol. The multipath striping protocol clocks the sending of its data at full link capacity and distributes the data by striping them along the two available paths. That is, given N packets destined for destination D, the protocol sends packet $2i$ on the one-hop path and packet $2i+1$ on the two-hop path to D, $0 \leq i < N/2$. The single path TCP protocol sends all packets along the

RED HAT 1012
RED HAT V. CAS

shortest path. In this experiment, nodes randomly select a neighbor and then send a burst of packets to that neighbor. The times between each burst are exponentially distributed. Figure 4.3 shows TCP throughput versus the striping protocol's throughput when all three nodes are sending data.

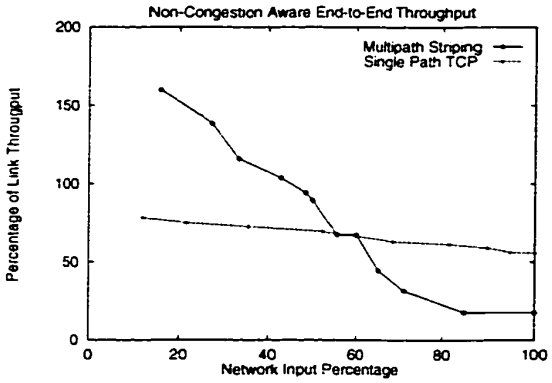


Figure 4.3 : Simulation result using the network in Figure 4.2. The graph shows the performance of single path TCP versus a multipath striping protocol without congestion control.

In Figure 4.3, the y-axis represents the average end-to-end throughput of all nodes as a percentage of link capacity, and the x-axis represents the aggregate sending rate normalized by the total network capacity.

First, the graph shows that TCP performance is very stable despite the increase in traffic. TCP does not achieve the maximum link bandwidth because its congestion and flow control mechanisms cautiously probes the network for available bandwidth and reduces its sending rate upon detection of congestion. On the other hand, the non-congestion aware multipath striping protocol achieves very high throughput at low network utilization levels. The reason is that at low network loads, the amount of contention from other connections is low; therefore, multiplexing data between the one link and two link paths allows effective aggregation of path resources, resulting in higher throughput than the single path strategy.

However, the relative performance of these two strategies changes at higher network loads ($\geq$ 60%): here, the TCP's transmission strategy proves superior because at these utilization levels, packets in the multipath striping protocol experience enough contention

RED HAT 1012
RED HAT V. CAS

from other connections to cause significant performance degradation (due to packet queuing in router buffers). This contention is due to packets traveling on the two link paths competing with packets from other connections. Furthermore, the contention increases as more packets are injected into the network, resulting in degradation of both aggregate and individual throughput.

To address this problem, this thesis develops a congestion-aware multipath transport protocol, MPTCP. Details of the MPTCP protocol are given in Chapter 7. The effectiveness of MPTCP is again compared against a single path TCP protocol on the same three node network. The results are shown in Figure 4.4.

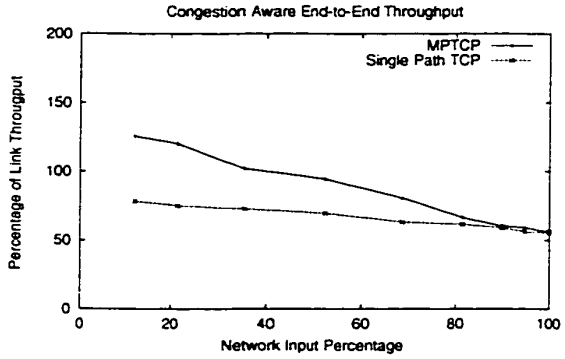


Figure 4.4 : Simulation result using the network in Figure 4.2. The graph shows the performance of single path TCP versus MPTCP.

The x-axis and y-axis have the same representation as the previous figure. As this figure shows, MPTCP outperforms the single path protocol at all levels of network utilization. This demonstrates MPTCP's ability to adapt to network conditions in order to increase throughput.

Compared to the multipath striping protocol, MPTCP does not achieve the same level of end-to-end throughput at low network utilization. This is because MPTCP is congestion aware and incrementally tests the network for available bandwidth. This cautious approach results in lower performance when the network is underutilized because it takes time for MPTCP to fully exploit available path bandwidth. However, the same cautious approach

RED HAT 1012
RED HAT V. CAS

allows MPTCP to significantly outperform the striping protocol at high utilization levels.

The conclusion of this section is that in order for end users to benefit from a multipath network, the network needs to provide not only the right paths, but the end-hosts also need to be able to take advantage of the additional paths. As shown by a naive multipath striping approach (Figure 4.3), wrongly using multiple paths can degrade not only end-to-end performance, but also the performance of other connections. This section also shows that correctly using multiple paths can increase network performance and avoid performance degradations.

4.2.5 Static and Dynamic Metric Multipath Routing

Like single path routing, multipath routing algorithms can use either dynamic or static metrics. In dynamic metric multipath routing, routing algorithms monitor link costs and recompute paths upon detection of link cost changes. This thesis does not specifically address the dynamic metric multipath routing model because we believe this model can be implemented by traditional dynamic metric triggering mechanisms; thus the methods presented in this thesis for static metric multipath routing are also applicable to dynamic metric multipath routing.

As stated in Chapter 2, the fundamental difference between multipath routing and single path dynamic metric routing is that end-hosts in multipath networks control the use of its paths on a much finer time and path granularity. Thus, given appropriate paths and end-host protocols, multipath end-hosts can dynamically detect poor path performance, and then switch and use other paths that provide better performance.

For example, consider an application that wishes to minimize its communication delay to its destination, and the multipath routing algorithm calculates link disjoint paths. In single path dynamic metric routing, if the path to the destination is congested, then the application will incur the delay caused by the congestion, unless the routing algorithm recompute a better path. In multipath routing, however, the application can dynamically switch and use other paths to avoid the congestion without any router intervention. Thus, on

RED HAT 1012
RED HAT V. CAS

small time scales, the ability to control path usage allows end-hosts in multipath networks to dynamically adjust to transient traffic patterns.[†]

Because end-hosts can adapt to small time-scale traffic patterns, a dynamic metric multipath routing algorithm does not need to recompute paths in fine time granularities. Consequently, these routing algorithms should recompute paths that consider large time-scale traffic patterns. For example, a dynamic metric multipath routing algorithm might monitor traffic patterns for weeks and then recompute paths based on the gathered statistics. This approach to path recomputation provides better paths by considering traffic trends rather than transient traffic bursts (which are addressed by end-host multipath protocols). On this time scale, problems such as route oscillations and excessive routing costs do not occur. Thus, the traditional LS and DV triggering mechanisms are sufficient to implement the dynamic multipath routing model. The multipath routing algorithms developed in this thesis are based on DV and LS.

4.3 Multipath Routing Summary

While it is clear that the static metric multipath routing model offers many advantages over single path routing models, it is unclear whether enough benefits could be extracted to offset the cost of their implementation. In short, in order to make multipath routing viable, the following questions need to be resolved:

1. What paths should be calculated between nodes and how?
2. How should routers efficiently provide multiple paths in a distributed routing environment?
3. How should end-hosts use multiple paths to gain higher performance?

[†]This conclusion assumes that the multipath routing algorithm provides quality paths between nodes (Chapter 5) and that end-hosts effectively use these paths to increase their performance (Chapter 7).

RED HAT 1012
RED HAT V. CAS

The first question deals with the potential gains of a multipath network. As illustrated in Section 4.2.4, one of the necessary criteria for multipath networks to increase end-to-end performance is to calculate the right paths. To address this issue, Chapter 5 surveys different multipath calculation algorithms. We then develop two algorithms, one maximizing throughput and the other minimizing delay. The two algorithms appear in Chapter 5.

The second question deals with the cost of providing multiple paths between nodes. The main cost of implementing multipath routing is solving the packet forwarding problem: how to efficiently forward packets to the same destination but on different paths? The novel solution developed in this thesis uses routing overhead linear in the number of paths between nodes and has constant per packet path specification overhead. This low overhead is achieved by requiring that paths calculated by a multipath routing algorithm satisfy the *suffix matched* property. Complete details of this forwarding method and its requirements are given in Chapter 6.

The last question is how end-hosts should best use a multipath network in order to increase their performance. As demonstrated in the previous section, the way in which multiple paths are used has a dramatic impact on individual and aggregate performance. A congestion aware multipath transport protocol, MPTCP, is developed that effectively uses multiple paths to increase throughput. The details of the protocol and its strategies are provided in Chapter 7.

The algorithms and methods developed in Chapters 5 – 7 are combined to implement a multipath network. Two multipath routing algorithms are implemented: one based on LS and the other on DV. Details of their implementation are given in Chapter 8. Finally, Chapter 9 presents simulation results that measure multipath routing's obtained performance and incurred costs.

RED HAT 1012
RED HAT V. CAS

Chapter 5

Path Calculation Algorithms

To provide multiple paths between nodes, a multipath routing algorithm first needs to calculate the paths it wishes to provide. A *multipath calculation algorithm* refers to an algorithm that calculates multiple paths between node pairs. The paths calculated between nodes directly affects the performance gains a node pair can obtain. For example, a throughput application can only increase its throughput if the paths provided has a larger aggregate bandwidth than the path provided by a single path routing algorithm.

To this end, this chapter surveys different path calculation algorithms and develops two multipath calculation algorithms, one that computes low delay paths and the other high bandwidth paths. Both algorithms are based on Dijkstra's shortest path algorithm.

Before presenting the two algorithms, we first specify two path characteristics that allow end-hosts to benefit from using multiple paths. Section 5.1 presents these two path characteristics. Then in the succeeding sections, we use these characteristics to guide the development of multipath calculation algorithms that calculate paths for low delay and high bandwidth path services.

5.1 Path Characteristics

The quality of a path set depends on the intended use of the paths in the set. For example, a path set that provides high throughput may be very different from one that provides low delay. Although path set quality (measured in potential performance improvements) depends on the intended use of the path set, in general, there are some path characteristics that allow applications to increase their performance.

In this section, we describe two such path characteristics, *path quantity* and *path inde-*

RED HAT 1012
RED HAT V. CAS

pendence. Path quantity refers to the number of paths calculated between nodes, and path independence refers to the disjointedness of the calculated paths. These two characteristics are based on the basic advantage of multipath routing – the increase of end-to-end performance by providing multiple paths between nodes. In general, this increase in performance is obtained through 1) aggregating path resources and 2) choosing the best path(s) among the available paths. We describe the two path characteristics below.

Path Quantity

The fundamental difference between a multipath network and a single path network is the number of paths provided between nodes. With more paths, nodes have more options to interact with other nodes, potentially increasing their performance. Thus, the higher the number of paths calculated between nodes, the higher potential a path set can improve end-to-end performance.

There are many ways to describe path quantity. One method is to consider the total number of paths calculated in a path set; the larger the number, the better the path set. While this is straightforward, it does not capture the distribution of paths between node pairs. For example, assuming uniform traffic distribution, a path set that provides every node pair with 3 paths should be better than one that provides half the node pairs with 1 path and the other half with 5. That is, path sets with the same averages but higher variance are less desirable because it means that some nodes have few paths while others have many.

Although there are subtleties in the precise description of path quantity, in general, a multipath calculation algorithm that calculates more paths is better than one that calculates less. However, path quantity only characterizes one aspect of a multipath set. Another characterization is the independence among paths calculated between a node pair. This is described next.

RED HAT 1012
RED HAT V. CAS

Path Independence

The second path characteristic, path independence, describes the disjointedness of the paths provided between nodes. This property is important because the more independent a path set, the more aggregate physical resources the set offers between a node pair (because those resources are not shared), and the less likely the performance of one path affects performances of other paths.

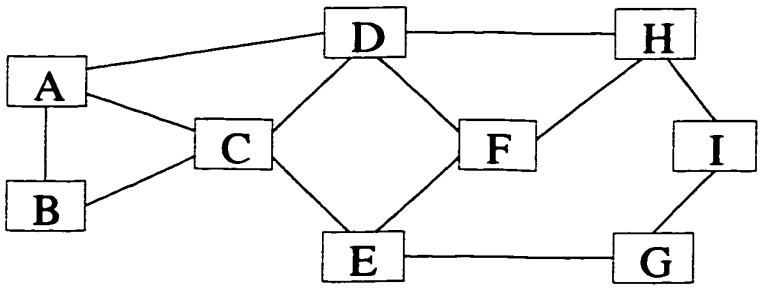


Figure 5.1 : An Example Network Topology. All links have uniform capacity and delay.

To illustrate the importance of path independence, consider the network in Figure 5.1. Assume that a path set that has two paths from A to I as (A, D, H, I) and (A, C, D, H, I) and another path set with paths (A, D, H, I) and (A, C, E, G, I). Intuitively, paths in the second set are more independent than the ones in the first set because paths in the second set do not share links. The higher independence of the second set gives node A more aggregate capacity to I. In addition, suppose a link on A's shorter path is congested, then the probability that A's other path to I is also congested is less likely in the second path set. This is because with the second set, at least one link in each paths must to be congested in order for A to suffer congestion; whereas in the first set, A would suffer congestion affects if either link (D, H) or (H, I) is congested.

In addition to increasing performance, path independence also allows end-hosts to achieve higher network availability. Assume that a network link has failed and that end-

RED HAT 1012
RED HAT V. CAS

hosts can detect that a path is not delivering data* faster than the routing algorithm. In this case, an end-host has a higher probability of reestablishing connectivity on a different path if the paths provided are independent. In contrast, in a single path environment, where there is no path independence (because there is only one path between nodes), an end-host must wait for the routing algorithm to detect the failure and recompute the appropriate paths in order for the end-host to reestablish connectivity.

In summary, we described two desirable multipath characteristics, path quantity and path independence. Multipath sets with these characteristics better allow end-hosts to increase their performance, compared with path sets that do not.

5.2 Path Calculation Algorithms

This section uses path quantity and independence to guide the development of multipath calculation algorithms for high throughput and low delay path services. These two services are chosen because their paths can provide immediate benefits to current network applications. The efficiency requirement of the considered algorithms is that their message and CPU usage are within a linear factor in the number of paths calculated, compared to computing single shortest paths. This complexity bound is imposed to make the implementation of these algorithms feasible in real networks.

In both low delay and high throughput paths, the performance of these paths depends on the dynamics of network traffic. As stated in Chapter 4, a multipath routing algorithm (both static and dynamic metrics) should compute paths that allow end-hosts to adapt to transient (fine time-scale) traffic dynamics. That is, the paths provided between nodes should be such that they preserve their intended performance benefits under varying traffic conditions.

We use the two path characteristics given in the previous section to guide the development of path calculation algorithms that compute paths which preserve their performance

*End-hosts can detect failed links in a path by observing that the path is not delivering data to the destination.

RED HAT 1012
RED HAT V. CAS

benefits under different traffic conditions. Section 5.2.1 presents an algorithm that computes paths to decrease transmission delay, and Section 5.2.2 presents an algorithm that increases throughput.

5.2.1 Minimizing Delay

One of the ways end-hosts can benefit from a multipath network is to lower their communication delays (or increase responsiveness) to other nodes. This can be done by monitoring the round trip delay of available paths to the desired destination and dynamically choosing the one with the least delay[†]. Since the minimal delay that an end-host can achieve depends on the delay of the paths it has to its destination, end-host delay performance is closely tied with the paths calculated. This section develops a path calculation algorithm that provides the low delay path service.

In order to calculate paths that minimize delay, a path calculation algorithm needs to gauge the expected delay of a link. For the following discussion, we assume that link delays are given by a cost function; the higher a link's cost, the higher the expected latency of the link. In addition, for ease of explanation, we assume uniform link costs; therefore, a path's delay is determined by the number of hops.

For the low delay path service, the objective is to calculate paths so that the smallest delay path between a node pair is minimized. Given that a path's actual delay is traffic dependent, a multipath calculation algorithm needs to calculate paths so that low delay can be obtained under different traffic patterns. Below, we present several potential algorithms and highlight their advantages and disadvantages.

[†]Path symmetry is not needed here. Path symmetry refers to the property that the path from A to B must be identical to the reverse of the path from B to A. Symmetry is not necessary in real networks because network traffic in one direction has no correlation with traffic behavior in the reverse. Thus, path symmetry is not required for the correct functioning of data transport protocols.

RED HAT 1012
RED HAT V. CAS

Shortest K Paths

The most natural algorithm for minimizing delay is one that calculates paths with the shortest total delay. A family of algorithms, called the shortest K algorithms, compute K paths such that the total cost of the paths is minimized. Moreover, there are efficient solutions to implement these algorithms [45, 58, 165].

Given that the link costs reflect delay, these algorithms provide path sets with the minimum total delay. However, the problem with these sets is that they do not consider path independence. For example, consider the network in Figure 5.1 where three paths are computed from node A to H. A shortest K algorithm will compute the paths (A, D, H), (A, C, D, H), and (A, D, F, H). Although this path set the three smallest hop paths (8 hops total), it is clearly not a desirable set because of the high degree of link sharing (i.e. lack of path independence). For example, in this path set, if the links on the shortest path are congested (i.e. links (A, D) and (D, H)), then the other two paths will be congested as well because these two links are also part of those paths. This lack of independence limits the amount of benefits end-hosts can obtain under different traffic conditions.

In general, with shortest K algorithms, paths between node pairs will tend to share many links, which reduces the effectiveness of providing multiple paths.

Link Disjoint Paths

Algorithms that overcome the problem of path independence are ones that calculate link disjoint paths between nodes. Link disjoint paths are paths that do not have any links in common. Like the shortest K algorithm, there are many algorithms to efficiently calculate these path sets [36, 118, 146, 152].

However, the link disjoint algorithm achieves high path independence at the expense of path quantity and path delay. For example, consider the paths calculated by a link disjoint algorithm from A to H in Figure 5.1. One possible path set is (A, D, H), (A, C, E, F, H), and (A, B, C, E, G, I, H). Here, the paths are disjoint, but the combined length of the path set is 12 hops. This is 50% longer than the path set calculated by shortest K algorithm.

RED HAT 1012
RED HAT V. CAS

Because different paths between a node pair cannot share links, this may be too restrictive, resulting in paths of excessive delay or a small number of calculated paths. Both of these consequences lowers the probability that end-hosts can obtain low delay paths to their destinations.

Discount shortest path

In considering path quantity and path independence, we developed a path calculation algorithm called the discount shortest path algorithm. The idea of the algorithm is to add a uniform cost to all links on the previously calculated paths. Therefore the paths calculated are compromises between the paths calculated by the shortest K and link disjoint algorithms: the link disjoint algorithm is one that adds an infinite cost to used links, while the shortest K algorithm selectively adds the minimum cost to the used links so that the next shortest path calculated is distinct from the computed paths. The basic discount shortest path algorithm is a variation of Dijkstra's shortest path algorithm. Similar variations have been published in the literature [156].

The discount shortest path algorithm assumes that for paths between any two nodes, there is an upper bound on the cost of the longest path. This is reasonable since most path calculation algorithms do not wish to calculate arbitrarily long paths. C_{max} is used to denote the maximum admissible path cost between a node pair. An example of C_{max} is 3 times the cost of the shortest path [94, 129].

The discount shortest path algorithm calculates paths with the following properties: from node a to b, the ith path is the least-cost path from a to b such that the path's cost is less than C_{max}. The cost of path i is calculated after adding cost increments to each link in path j from a to b, $1 \leq j < i$, where the cost increment of a path P is $(C_{max} - Cost(P))/Length(P)$.

To calculate discount shortest paths from node a to b, the algorithm first calculates the shortest path P with cost C_p from a to b. C_{max} is then calculated as `Cost_BOUND`$*C_p$ (in our implementation, `Cost_BOUND` is set to 3). Next, the cost increment for this path

RED HAT 1012
RED HAT V. CAS

is calculated as $P_{incr} = (C_{max} + 1) - C_p$. That is, a path's cost is incremented by the smallest amount so that the path exceeds the C_{max} and therefore will not be admissible in subsequent computations. This cost increment is then added uniformly to the cost of all links on path P. That is, for every link in P, the cost is incremented by $P_{incr}/length(P)$.

To get the next path from a to b, this process is repeated using the newly incremented link costs. The algorithm stops when either K paths are computed or there does not exist paths from a to b with cost less than C_{max}. After computing the paths from a to b, the link costs are restored to their original costs, and the discount shortest path computation begins for another node pair. The pseudocode for the discount shortest path algorithm is given in Figure 5.2.

To calculate K paths from node a to b, the discount shortest path algorithm iterates Figure 5.2's main loop K times. In each iteration, the function `Get_Shortest_Path()` is called. Given E edges and n nodes, the function which takes $O(E * lg(E))$. In addition, on each iteration, each link in the newly calculated path is traversed, which takes $O(n)$. At the end of the loop, the added link costs are restored ($O(K * n)$). Thus, the complexity of the discount shortest path algorithm in computing K paths from a to b is $O(K * (E * lg(E) + n) + K * n) \rightarrow O(K * E * lg(E))$. Notice that this is K times the complexity of calculating the single shortest path between two nodes.

5.2.2 Maximizing Throughput

This subsection describes a path calculation algorithm for another path service, the throughput path service. Throughput oriented applications such as FTP benefit from this path service because multiple paths can increase their effective throughput. Again, the amount of throughput an application obtains depends not only on the paths calculated, but also on network traffic patterns. Thus, the paths calculated should be robust in the sense that they provide high throughput under a variety of traffic conditions.

Again, without loss of generality, we assume that expected link bandwidth can be characterized by a link capacity metric; the higher the link capacity, the higher the expected

RED HAT 1012
RED HAT V. CAS

```
numPaths = 0;
while(numPaths < K)
{
    newPath = Get_Shortest_Path(Src, Dst);
    if (newPath == NULL)
        break;
    if (numPath == 0)
        Max_cost = Cost(newPath) * Cost_BOUND;
    if (Cost(newPath) > Max_cost)
        break;
    numPaths++;
    StorePath(Src, Dst, newPath);
    Cost_diff = Max_cost - Cost(newPath) +1;
    Cost_incr = Cost_diff / Length(newPath);
    forall links l ∈ newPath
        l.cost = l.cost + Cost_incr;
}
Restore all link cost additions
```

Figure 5.2 : The pseudocode for the discount shortest path algorithm. The code shows the calculation of K discount shortest paths between Src and Dst.

RED HAT 1012
RED HAT V. CAS

available bandwidth of the link.

Maximum Flow Paths

The most straightforward algorithms to calculate throughput paths are maximum flow algorithms. Calculating a set of links that provides the maximum flow between two nodes can be done using conventional flow algorithms [5,42]. However, applying these algorithms directly has two main drawbacks.

First, the maximum flow algorithms do not produce a set of end-to-end paths; that is, a maximum flow algorithm produces a set of links and capacities so that sending the specified amount of data on each of those links provides the maximum flow. Thus, to calculate paths requires an additional level of processing to convert the set of links into a set of paths. In addition, since routing algorithms typically bound the number of paths it calculates (say K) between a node pair, an algorithm needs to select K paths that maximizes flow from the set of links. Selecting these K paths is not trivial because it depends on how the paths are initially chosen from the maximum flow link set.

The second drawback of directly apply maximum flow algorithms is that these algorithms do not consider path length (or path costs). Given that routing algorithms typically want to control the length of the paths they calculate, the paths derived from a maximum flow calculation may not provide a good path set given a length threshold.

Capacity Removal Algorithm

Due to the drawbacks of strictly applying maximum flow algorithms, we have developed an algorithm called the capacity removal algorithm based on Dijkstra's shortest path algorithm. Similar to maximum flow, this algorithm calculates paths that aim to increase the flow between node pairs. Moreover, the capacity removal algorithm explicitly considers the number of paths to calculate and path length.

As in the discount shortest path algorithm, the capacity removal algorithm calculates successive shortest paths; after calculating a path, the algorithm subtracts the path capacity

RED HAT 1012
RED HAT V. CAS

from every link along that path. The capacity of a path is the minimal capacity of all links on the path. A link capacity threshold is used so that links with capacities below the threshold are eliminated from subsequent path computations. The pseudocode for the capacity removal algorithm is given in Figure 5.3.

The description of capacity removal paths can be summarized as follows: from node a to b, the ith path is the least-cost path from a to b with cost less than C_{max} and capacity greater than the capacity threshold, where path i's capacity is calculated after subtracting the path j's capacity from every link in path j, $1 \leq j < i$.

The complexity analysis of the capacity removal algorithm is very similar to the discount shortest path algorithm. To calculate K paths from node a to b, the capacity removal algorithm iterates Figure 5.3's main loop K times. In each iteration, the function `Get_Shortest_Path_CapThresh()` is called which takes $O(E * lg(E))$, and each link in the newly calculated path is traversed $O(n)$ times. At the end of the loop, link capacities are restored ($O(K * n)$). Thus, the complexity of the capacity removal algorithm to calculate K paths between a node pair is $O(K*(E*lg(E)+n)+K*n) \rightarrow O(K*E*lg(E))$. Again, like discount shortest path, the complexity of the capacity removal algorithm to calculate K paths is K times the complexity of calculating the single shortest path between two nodes.

5.3 Path Calculation Summary

This chapter presents two path calculation algorithms, one computing paths for low delay path service and the other for high throughput path service. The two algorithms developed, discount shortest path and capacity removal, are both shortest path based algorithms, and their complexities are linear in the number of paths calculated compared to computing single shortest paths.

The quality of capacity removal paths, measured in achievable throughput, is evaluated in Chapters 7 and 9. Simulation results show that the capacity removal algorithm effectively provides paths that allow end-hosts to increase their throughput.

RED HAT 1012
RED HAT V. CAS

```
numPaths = 0;
while(numPaths < K)
{
    newPath = Get_Shortest_Path_CapThresh(Src, Dst,
                                    Capacity_Threshold);
    if (newPath == NULL)
        break;
    if (numPath == 0)
        Max_cost = Cost(newPath) * Cost_BOUND;
    if (Cost(newPath) > Max_cost)
        break;
    numPaths++;
    StorePath(Src, Dst, newPath);
    Path_cap = Capacity(newPath);
    forall links l ∈ newPath
        l.capacity = l.capacity - Path_cap;
}
Restore all link capacity subtractions
```

Figure 5.3 : The pseudocode for the capacity removal algorithm. The code shows the calculation of K capacity removal paths between Src and Dst. The function `Get_Shortest_Path_CapThresh(Src, Dst, Cap_thresh)` returns the shortest path from `Src` to `Dst` such that all links in the path have capacity above `Cap_thresh`.

RED HAT 1012
RED HAT V. CAS

Chapter 6

Multipath Forwarding

Network routing is a two-step process: the first step is computing the desired paths, and the second is forwarding data on those paths. In a multipath network, the cost required to support these two steps is higher than in single path routing. A summary of the multipath overheads are

1. Algorithmic cost: router CPU and memory required to calculate multiple paths
2. Routing costs: extra forwarding table entries, CPU cycles, and routing messages
3. Per packet path specification costs: number of bits required to specify a packet's path
4. Per packet forwarding time

For all but the algorithmic cost (addressed in Chapter 5), the costs listed above are directly related to how paths are encoded and forwarded in a network. This chapter presents solutions that efficiently accomplishes this forwarding task.

In multipath networks, packets need to specify not only their destinations, but a specific path to their destinations. This means that routers need to recognize path specifications (or encodings) in order to forward packets correctly. The problem of encoding and forwarding packets along their intended path is called the *path forwarding problem.*

Methods of solving the path forwarding problem depend on whether packets are forwarded on paths within the same path service (multi-option paths) or on paths from different path services (multi-service paths). This differentiation is important because it affects the implementation of the packet forwarding method. Path forwarding on different service paths can be implemented in a straightforward manner using a service identifier; however,

RED HAT 1012
RED HAT V. CAS

this encoding scheme is not sufficient for multi-option path forwarding. Because multi-service forwarding is relatively straightforward, this chapter briefly discusses multi-service forwarding but primarily focuses on multi-option forwarding.

This chapter develops an efficient solution for the multi-option forwarding problem. A forwarding method is efficient if the per packet path specification cost is small (in the number of bits) and the path specification allows fast router packet forwarding. The solution developed uses 1) a per packet overhead of a small, fixed-length path identifier, and 2) router space overhead linear in K, the number of paths calculated between nodes. To achieve these efficient bounds, the forwarding method requires that multi-option path sets satisfy the *suffix matched* property.

The remainder of this chapter is organized as follows. The following section formally defines the multipath forwarding problem, both multi-service and multi-option. The section describes multi-service forwarding and develops the basis the proposed multi-option forwarding method. In Sections 6.2 and 6.3, the multi-option forwarding method is applied to Distance Vector and Link State routing algorithms. In addition, this chapter contains proofs that 1) Distance Vector based algorithms compute suffix matched multipath sets, and 2) Link State based algorithms yield suffix matched multipath sets for the criterion of ranked k-shortest paths. Section 6.4 provides an example to demonstrate the forwarding method in the context of a multi-service multi-option network. Finally, a summary of the path forwarding method appears in Section 6.5.

6.1 The Multipath Forwarding Problem

A network can be represented as a graph $G = (V, E)$, where V is the set of network nodes and E is the set of links or edges. Each link $(x_i, x_j) \in E$ has an associated positive cost $c_{x_i x_j}$. A path p in G is a list of nodes $(x_1, \ldots, x_n)$ such that $\forall i, 1 \leq i < n$, $(x_i, x_{i+1}) \in E$. Path p has cost $\sum_{i=1}^{n-1} c_{x_i x_{i+1}}$. No node appears more than once in a *simple* or loop-free path.

Abstractly, to route packets amongst nodes in a network, each node $x \in V$ implements

RED HAT 1012
RED HAT V. CAS

two functions:

$$h_x : V \rightarrow \Lambda$$

$$G_x : \Lambda \rightarrow V$$

The *path selection* function h_x chooses a path identifier ψ (on a per-packet or per-connection basis) for a given destination node $d \in V$ from the set Λ. Path identifiers encode distinct paths from x to d. The forwarding function G_x takes a path identifier $\psi \in \Lambda$ and provides the next-hop neighbor on path ψ. As an example, if node x wishes to send a packet to node d on path ψ, it adds the label ψ to the packet, and forwards it to $G_x(\psi)$: the next-hop on path ψ from x to d.

Single shortest path routing is a special case where there is no need to have a path identifier set Λ. In this path setting, the path selection function is the identity function: $h_x(d) = d,\ d \in V$. This is because exactly one path p from x to d is used at any given time — there is no choice of paths.* By the Bellman optimality principle, the shortest path from any intermediate node n in p to node d is the subpath of p from n to d; thus from any node, the path to a destination d can uniquely identify by d's address.

One can replace the path identifier argument by the destination and simplify the forwarding function in a single shortest path setting to be $G_x : V \rightarrow V$ which identifies the next-hop for a given destination. Router x implements G_x by looking up its forwarding table.

In a multipath setting, an originating node x has a choice of several paths to a particular destination at any given time (i.e, the range of the function h_x is a set of size ≥ 1). The particular path (identified by $h_x(d) = \psi$ chosen by x for a destination d is a function of the sending host's objectives (refer to Section 5.2).

*The actual identity of the shortest path between two nodes in the network may change as the network changes, but at any given time a node can route to a destination along exactly one path.

RED HAT 1012
RED HAT V. CAS

The forwarding functions G_x, for each $x \in V$ need to respect the following constraint:

$\forall \psi \in \Lambda$ such that ψ identifies path $= (x_1, \ldots x_n)$,
$G_{x_i}(\psi) = x_{i+1}, 1 \leq i < n$.

Efficiently enforcing the above constraint on the G_x functions forms the crux of the multipath forwarding problem.

In the proposed multipath routing model, the forwarding problem can be divided into two parts, forwarding on different path services and forwarding within the same service. This division is possible because the way a packet specifies its path service is independent of how it specifies a path within that service. Thus, ψ can be represented by a fixed-length triple (Dst, ψ_1, ψ_2) where Dst is the destination address, ψ_1 specifies the path service, and ψ_2 specifies the particular path within the path service. This thesis uses this path identifier representation to uniquely identifies a particular path in a multipath network.

The following subsections use this three tuple ψ representation to address the multipath forwarding problem. The next subsection describes path forwarding on different path services (specifying ψ_1), and subsection 6.1.2 focuses on path forwarding among paths within the same service (specifying ψ_2).

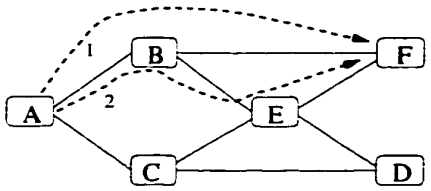
6.1.1 Multi-Service Path Forwarding

Forwarding packets on different path services can be accomplished by tagging each packet with a path service identifier. A service ID is an integer that distinguishes one path service from another. Because this identifier disambiguates packets from different services, the forwarding function G can be implemented by switching on service identifiers. That is, upon receiving a packet, a router forwards the packet using the forwarding function G specified by the packet's service ID.

For example, in a multi-service single-option network, the forwarding function G for

RED HAT 1012
RED HAT V. CAS

each service is the same as a single, shortest path forwarding function†. In this scenario, upon receiving a packet, a router simply forwards the packet to the next-hop returned by applying the function G specified by the path service.



A's forwarding table to F

service type	next-hop
1	B
2	B

B's forwarding Table to F

service type	next-hop
1	F
2	E

E's forwarding table to F

service type	next-hop
1	F
2	F

Figure 6.1 : A forwarding example in a multi-service single-option multipath network.

Figure 6.1 shows selected forwarding tables of a multi-service single-option network. Here, the forwarding tables of routers A, B, and E show that each router computes two service paths to F. The dashed lines show A's two paths to F; the number above the lines show their path service number. In this setting, the forwarding function guarantees that if A sends a packet to F and tags the packet with the intended path service number, the packet will travel the intended path to F.

For example, assume that A sends a packet on path service 2 to F. This packet is than tagged with the path identifier $[F, 2]$ and forwarded, according to A's forwarding table, to node B (for simplicity, the multi-option identifier ψ_2 is omitted because there is only one path per service). Upon receiving this packet, B looks up its forwarding table for destination F with service 2 and forwards the packet to node E. E performs the same lookup function and forwards the packet to F.

Path forwarding in this scenario is guaranteed because 1) the packet's service identifier ensures that every router uses the right forwarding function (e.g. looks up the appropriate

†This example uses single-option paths because we established earlier that the single shortest path forwarding function G correctly forwards packets because the paths satisfy the Bellman optimality principle.

RED HAT 1012
RED HAT V. CAS

forwarding table entry), and 2) because one path is calculated between nodes within each service, the single path forwarding function corresponding to each service guarantees that packets are forwarded on their specified paths.

As this example shows, multi-service paths can be distinguished using a simple path service identifier. The next section shows that a similar straight-forward multi-option identifier is not sufficient to guarantee that packets are correctly forwarded on multi-option paths.

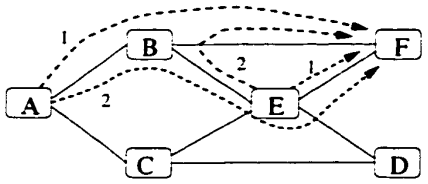
6.1.2 Multi-Option Path Forwarding

With multi-option paths, a router calculates multiple paths for the same path service. For the purpose of discussion, we assume that each multi-option path is ranked. That is, when a router computes multi-option paths to a destination, it locally assigns a unique number i to each multi-option path, indicating that the path is the ith path to that destination for a particular path service. For example, the ranking of paths could reflect the ith best path the router calculates within a path service.

Unlike multi-service forwarding, path forwarding for multi-option paths cannot be solved by simply tagging packets with the path's rank number. Because multi-service IDs are consistent and understood by all routers to denote a specific path service, tagging packets with a service ID unambiguously identifies a unique path service. In contrast, tagging a packet with the rank of a multi-option path, in general, does not guarantee that the packet will be forwarded on the specified path because multi-option ranks are not necessarily consistent in all routers.[‡] For example, assume that the path $(x_0, \ldots, x_n)$ is the 2nd best path from x_0 to x_n. It is not guaranteed that $\forall x_i, 0 \leq i \leq n$, x_i's 2nd best path is $(x_i, \ldots, x_n)$. Figure 6.2 illustrates this property.

[‡]If the multi-option IDs are consistent in all routers, then tagging packets with these IDs ensures correct path forwarding. However, making multi-option IDs consistent for a particular path calculation algorithm requires routers to know the paths computed by other routers, which increases both router computation and storage overheads. The Compute All method described in Section 6.1.3 is one such approach.

RED HAT 1012
RED HAT V. CAS



A's forwarding table to F

path number	next-hop
1	B
2	B

B's forwarding Table to F

path number	next-hop
1	F
2	E

E's forwarding table to F

path number	next-hop
1	F
2	B

Figure 6.2 : A forwarding example in a single-service multi-option multipath network.

In Figure 6.2, routers compute one path service with two multi-option paths, where the first path denotes the shortest path and the second denotes the second shortest path. The dashed lines show A and E's paths to node F. The figure demonstrates that a path number (or multi-option rank number) is not sufficient to ensure path forwarding. For example, if A wishes to send a packet on its second path to F and tags the packet with only the path number (i.e. path ID $[F, 2]$), the packet will *not* travel the intended path. To see this, after A sends the packet to B tagged with $[F, 2]$, B receives this packet and will forward the packet to B's second path's next-hop, E. E then forwards the packet on its second path, which has B as the next-hop. Notice that E *should* forward the packet on its first path (to node F). Following the example, B will then forward the packet back to E because E is the next-hop of B's second path. This results in the packet bouncing between E and B.

This example shows that because multi-option path ranks are not necessarily consistent in all routers, multi-option forwarding is not always guaranteed by simply tagging and forwarding packets based on rank numbers. To address this problem, the remainder of this chapter develops an efficient method to forward packets on multi-option paths. The next subsection describes suffix matched path sets, a crucial piece of the proposed method.

6.1.3 Suffix Matched Path Sets

This subsection presents an important class of path sets on which efficient multi-option forwarding can be performed. However, before defining this set, we first survey some

RED HAT 1012
RED HAT V. CAS

existing multipath forwarding methods.

To implement the G functions in a multi-option environment, one needs a scheme for constructing path identifiers that unambiguously identify paths between nodes. One common approach, called source routing, is to use the path description itself as the identifier. In this method, path identifiers are of variable length; therefore the overhead of tagging individual packets with these path IDs increases as the size of the network grows and the path length between nodes increases. The implementation of the G functions at each node x_i for path $(x_1, \ldots, x_n)$, $1 \leq i < n$, requires reading the received packet's path ID $[x_1, \ldots, x_n]$ and then computing $G_{x_i}([x_1, \ldots, x_n])$ to be x_{i+1}. No state information (e.g. forwarding table) is needed at intermediate nodes; however complete path information is needed at the sending nodes.

Although source routing is a general and flexible forwarding method, it is inefficient because of its variable length, per packet path ID: the variable length path ID increases the per packet path specification overhead, which decreases router forwarding efficiency because routers have to examine a larger packet header to determine the next-hop. In addition, source routing requires that sending nodes know the source routes of every path they wish to use, thereby increases router storage requirements. This chapter focuses on efficient path forwarding methods that use fixed-length path IDs and do not require sending nodes to know path source routes.

Another approach to multi-option forwarding is to establish a consistent set of multi-option IDs. The Compute All method is one such approach. With this approach, if K multi-option paths are maintained between source and destination pairs in an N node network, Compute All uniquely identifies a path by the triple (s, d, i), where $s, d, \in V$ denoting source and destination, and i is an integer, $1 \leq i \leq K$. To obtain these consistent IDs, each router computes, for each destination, not only its K paths, but every other router's K paths as well.

The aggregate forwarding table requirement for the Compute All method is $O(KN^2 * N) = O(KN^3)$, which is the size of the mapping from each path identifier to the next-

RED HAT 1012
RED HAT V. CAS

hop (there are KN^2 paths) and is maintained by each node (there are N nodes). This requirement can be reduced by observing that a node only needs to maintain paths that pass through it. Let L be the average path length, then each path passes through L nodes on average. Therefore it suffices that each node maintains only KNL path identifiers on average, reducing the total space requirements on forwarding tables to $O(KN^2L)$. In contrast, single shortest path routing only require an aggregating forwarding tables space complexity of $O(N^2)$.

Is it possible to efficiently forward packets on multi-option paths where packets are annotated by short fixed-length path identifiers with space overhead for forwarding tables no more than $O(KN^2)$, which is K times the cost for forwarding in single shortest path systems? This question is answered in the affirmative for a interesting class of multipath sets called *suffix matched* path sets.

Suffix matched path sets: A path set P is *suffix matched* iff for all paths $(x_1, \ldots, x_n) \in P$, then $\forall i, 1 \leq i < n, (x_i, \ldots, x_n)$ is also in P.

Consider the path set P consisting of the single shortest paths between every pair of nodes in a network. The Bellman optimality principle ensures that P is suffix matched. If the shortest path p from node x_1 to x_n is $(x_1, \ldots, x_n)$, then for $1 < i \leq n$, the shortest path from x_i to x_n is the subpath $(x_i, \ldots, x_n)$ of p.

Proposition: Packets in an N node network with a suffix matched multipath set P can be forwarded correctly with forwarding table space $O(KN^2)$ where no more than K paths are maintained between any pair of nodes.

Proof: Let the path identifier for path $p = (x_1, \ldots, x_n)$ from x_1 to x_n in a suffix matched multipath set P be $f_{x_1}(x_n)$. Let the path identifier for the suffix paths $p_i = (x_i, \ldots, x_n)$ for $1 < i \leq n$ be $f_{x_i}(x_n)$. Recall that all suffixes of p are in P by the definition of suffix matched multipath sets.

To implement path forwarding correctly, one needs to ensure node x_i knows the suffix of its path $f_{x_i}(x_n)$ (the path $(x_{i+1}, \ldots, x_n)$) at its neighbor x_{i+1} is called $f_{x_{i+1}}(x_n)$. Therefore one needs an identifier translation or swapping function $\mathrm{F}_{x_i} : \Lambda \rightarrow \Lambda$ at each node that

RED HAT 1012
RED HAT V. CAS

maps the path identifier $f_{x_i}(x_n)$ of suffix path p_i to the path identifier $f_{x_{i+1}}(x_n)$ for $1 \leq i \leq n$. That is

$$f_{x_{i+1}}(x_n) = \mathrm{F}_{x_i}(f_{x_i}(x_n)) \text{ and}$$

$$x_{i+1} = \mathrm{G}_{x_i}(f_{x_i}(x_n)).$$

In other words, when node x_i receives a packet with identifier $f_{x_i}(x_n)$ from its neighbor x_{i-1}, it forwards it to node $x_{i+1} = \mathrm{G}_{x_i}(f_{x_i}(x_n))$ along with the new identifier $f_{x+1}(x_n) = \mathrm{F}_{x_i}(f_{x_i}(x_n))$. Each x_j , $i < j < n$, performs the same two operations until the packet reaches x_n.

Notice that the definition of a suffix matched path set in a distributed environment implies that if router x_0 computes the path $(x_0, \ldots, x_n)$, then router x_i must also compute the path $(x_i, \ldots, x_n)$, $\forall i, 0 < i < n$. This ensures that there exists a F_{x_i} implemented by x_i that is guaranteed to translate a path identifier of x_i to x_{i+1}'s corresponding path identifier.

The space requirements for the implementation of the mapping F is proportional to the total number of paths in the network, i.e. $O(K'N^2)$. This is because each path is matched with at most one other path (there are $O(K'N^2)$ paths), therefore the matching table is at most $O(2K'N^2)$. In addition, the space requirement for the mapping G is simply the number of paths $O(K'N^2)$. Therefore multipath forwarding of suffix matched paths can be implemented with an aggregate space requirement of $O(K'N^2)$ QED.

In the transient state when suffix matched path sets are in the process of being computed at each router, correct path forwarding is not guaranteed because transient path sets are not guaranteed to be suffix matched. In fact, this absence of forwarding guarantees during route transitions affects most distributed routing algorithms, including Distance Vector and Link State based algorithms.

6.2 Distance Vector Extension

This section describes a multi-option extension to a Distance Vector (DV) based routing algorithm. Multiple services are not specifically addressed here because path forwarding

RED HAT 1012
RED HAT V. CAS

between services can be accomplished with a simple path service ID. The description of the basic DV algorithm is given in Section 2.1. The next section extends the basic DV algorithm to compute multi-option paths. We then prove that the new multipath DV algorithm computes suffix matched paths sets. The proposed forwarding method is applied to the extended DV algorithm in Section 6.2.2. The computation and space complexities are also given in this section. Finally, an example of the DV suffix matched forwarding method is presented in Section 6.2.3

6.2.1 Methods of Calculating Multiple Paths in DV

The single shortest path DV algorithm can be extended to allow the calculation of multiple paths between each pair of nodes. For instance, to calculate the ranked k shortest paths to each destination, each router maintains in its forwarding table (and advertises in its DVPs) k entries for each destination. Upon receiving DVPs from its m neighbors, the router computes its own k best paths from possible $m * k$ paths received from its neighbors.

Path selection criteria other than k ranked paths, such as link or node disjoint paths, can also be implemented in a DV environment. We show that any DV-based multipath routing algorithm that satisfies the following conditions calculates suffix matched path sets.

1. A router r may install a forwarding table entry with $N_d^r = s$ only upon receiving a corresponding distance vector for destination d from its neighbor s.

2. A router r never advertises a distance vector for destination d unless it has a corresponding forwarding table entry for d.

Proposition: The path set P calculated by a multipath DV routing algorithm that satisfies conditions 1 and 2 are suffix matched.

Proof: Assume the algorithm calculates a path $(x_0, \ldots, x_n) \in P$ that is not suffix matched. Then there must exist a router x_i, $0 < i < n$ such that $(x_{i-1}, \ldots, x_n) \in P$, but $(x_i, \ldots, x_n) \notin P$. That is, x_{i-1} has a forwarding table entry with $N_d^{x_{i-1}} = x_i$, but x_i does *not* have a forwarding table entry with $N_d^{x_i} = x_{i+1}$. Thus, either x_{i-1} has installed the forwarding table

RED HAT 1012
RED HAT V. CAS

entry without receiving a corresponding distance vector from x_i, or x_i has advertised a distance vector without having a corresponding forwarding table entry. But, these possibilities are ruled out by conditions 1 and 2 QED.

6.2.2 Multipath DV Extensions

In this section, we show how the proposed multipath forwarding method presented in Section 6.1 can be applied to any DV based multipath algorithms that satisfies conditions 1 and 2, and thus calculates suffix matched path sets.

The implementation of functions F and G, defined in Section 6.1, in a DV environment is defined as follows. Each router assigns an identifier ϕ for each of its paths to a given destination. Let $\phi^k_{r,d}$ be the ID router r assigns its kth path to destination d. Then, the path ID used by router r in the MPDV algorithm is $pid = [d, \phi^k_{r,d}]$[§]. Adding ϕ and evaluating F and G during the packet forwarding process requires the following additions to the DV data structures.

- Forwarding table entries have two additional elements, ϕ_1 and ϕ_2. An entry in router r's forwarding table for the path $(r, s, \ldots, d)$ is now a five tuple $(d, C^r_{d,\phi}, s, \phi_1, \phi_2)$. ϕ_1 is r's identifier for the path $(r, s, \ldots, d)$ and ϕ_2 is neighbor s's identifier for the corresponding suffix path $(s, \ldots, d)$. Note that $N^r_d = s$. $C^r_{d,\phi}$ is the cost of r's ϕ path to d.

- DVP entries have one additional element, ϕ. A DVP originating from router r that advertises r's path to destination d with identifier ϕ_1 is now a three tuple $(d, C^r_{d,\phi_1}, \phi_1)$.

Let D be the set of destinations and Φ^r_d be r's set of path identifiers to destination d. In its DVP, each router r advertises a distance vector $(d, C^r_{d,\phi_1}, \phi_1)$ corresponding to its forwarding table entry $(d, C^r_{d,\phi_1}, s, \phi_1, \phi_2)$, for each $d \in D$ and for each $\phi_1 \in \Phi^r_d$.

[§]To simplify the explanation, the path service identifier ψ_1 is omitted here. An example later in the chapter incorporates both path specifiers.

RED HAT 1012
RED HAT V. CAS

To install a route in its forwarding table upon receiving a distance vector $(d, C^s_{d,\phi}, \phi)$ from s, router r adds the entry $(d, C^r_{d,\phi_1}, s, \phi_1, \phi)$, where ϕ_1 is chosen by r to be unique in Φ^r_d; i.e. unique in its set of path identifiers to destination d. For instance, in the DV multipath algorithm for k ranked paths, r's identifier for the ith path to destination d can be chosen as i. C^r_{d,ϕ_1}, the cost of this path from s to d, is calculated in the usual manner as $C^s_{d,\phi_1} + c_{rs}$.

The functions F and G can now be implemented directly via lookup of the forwarding table. For path ID $[d, \alpha]$, node r evaluates $F_r([d, \alpha])$ as $[d, \beta]$ via lookup of the forwarding table entry for destination d and local identifier $\phi_1 = \alpha$, with $\beta = \phi_2$. Similarly, r evaluates $G_r([d, \alpha])$ as $N^r_{d,\alpha}$ via lookup of the same forwarding table entry. A matching forwarding entry is guaranteed to exist when evaluating F and G because the path sets are suffix matched.

The additional storage required for the MPDV forwarding method is two path identifier fields in each forwarding table entry and one in each DVP entry. Let K be the maximal number of paths calculated to each destination, then the storage required for path identifiers is $\lceil(\log_2(K))\rceil$ bits. The additional entries in the forwarding tables and DVP for MPDV correspond to the number of alternate paths being calculated. That is, the storage overhead is a factor of $O(K)$ more than single shortest path DV. The aggregate router storage complexity of MPDV is therefore $O(KN^2)$.

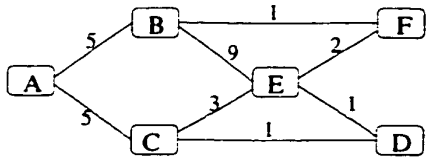
6.2.3 MPDV Example

The graph in Figure 6.3 represents a network and the number above each link represents the cost of that link[¶]. In table below are forwarding tables of selected routers to node F. The forwarding tables are computed using the Distance Vector multipath extension algorithm. For illustrative purposes, the path calculation algorithm calculates all paths with cost less than 10. Notice that this path calculation criteria does not exclude paths that traverse a node

[¶]For simplicity, this example assumes that link costs are symmetric. MPDV functions correctly with non-symmetric link costs as well.

RED HAT 1012
RED HAT V. CAS

more than once. The number of paths to any destination is at most 4, ie. $K = 4$. This DV version is assumed to implement split-horizon. Below, we describe the operation of this MPDV algorithm in two phases: path calculation/forwarding table construction and packet forwarding.



E's forwarding table to F

cost	Φ_1	Φ_2	next-hop
2	1	1	F
10	2	1	B
7	3	2	D
7	4	2	C

D's forwarding Table to F

cost	Φ_1	Φ_2	next-hop
3	1	1	E
6	2	2	C
8	3	4	E

C's forwarding table to F

cost	Φ_1	Φ_2	next-hop
5	1	1	E
4	2	1	D
10	3	3	E
9	4	3	D

Figure 6.3 : Example of MPDV.

Forwarding table construction. The process of calculating paths to node F begins by F sending a DVP to its neighbors (E and B). The DVP contains the path (F, 0, 1), where F is the destination address, 0 is the cost for F to reach F, and $1 = \phi^1_{F,F}$. When neighbor E receives this DVP, E adds the cost $C_{EF} = 2$ to every element in F's DVP. Since the path to F has cost $2 = 0 + 2$, it satisfies the path calculation criteria. E keeps this path and augments its forwarding table with the entry (F, 2, 1, 1, F): F is the destination address, 2 is the cost of this path to F, the first $1 = \phi^1_{E,F}$ (E's name for this path to F), the second $1 = \phi^1_{F,F}$ (node F's name for this path taken from F's DVP), and $F = N^E_{F,\phi^1_{F,F}}$ is this path's next-hop neighbor.

Next time E sends out its DVP, it will contain the path entry (F, 2, 1) advertising a path to F with cost 2 and $\phi^1_{E,F}$ of 1. When neighbor C receives this DVP containing this path to F, C augments its forwarding table with the path (F, 5, 1, 1, E) because this path is admissible under the path calculation criteria. In this example, C also chose the name of this path to be $\phi = 1$. This process of path computation continues until all routers have admissible paths to all other routers. As in the single path DV algorithm, the convergence time of MPDV is proportional to the longest path calculated between any two nodes. The

RED HAT 1012
RED HAT V. CAS

complete forwarding tables to F for routers E, D, and C are shown in Figure 6.3.

Packet forwarding. Suppose node D wishes to send a packet to node F and D decides to send on its third best path, the one with cost 8. According to the algorithm, D tags the packet with the path ID $[F, 4]$, where F is the destination address and $4 = \phi_2$ in the forwarding entry, and forwards the packet to the next-hop neighbor E. On receiving this packet, E looks up its forwarding table and matches F and ϕ_1 with the packet's path ID. The matching entry is the last entry in E's forwarding table. Once found, E replaces the packet's path ID with [F, 2] (E's ϕ_2 for this path is 2) and forwards the packet to C. That is $\mathrm{F}_E([F,4]) = [F,2]$ and $\mathrm{G}_E([F,4]) = C$. C does the same lookup and forwards the packet back to D with path ID $[F, 1]$ (here $\mathrm{F}_C([F,2]) = [F,1]$ and $\mathrm{G}_C([F,2]) = D$). Following the algorithm, D again performs the lookup and forwards the packet to E with the path ID $[F, 1]$. On receiving this packet, E finally forwards the packet to F with the ID $[F, 1]$.

For simplicity, the MPDV path calculation algorithm presented in this example did not eliminate finite looping paths. Since this path forwarding method ensures forwarding on all calculated paths, the algorithm will forward packets on finite-looping paths (e.g. path (D, E, C, D, E, F)) as well as non-looping paths (e.g. (D, E, F)). There are well known methods of ensuring loop-free path calculation in distance vector; refer to [93] for details.

6.3 Link State Extension

This section extends Link State (LS) based routing algorithms to use the proposed multi-option forwarding method. The LS algorithm, like DV, is widely used and well understood. However, unlike DV, it relies on a centralized route calculation algorithm. A topology broadcast mechanism ensures that each router knows the current state of the entire network (i.e. topology and link cost), and routes are calculated in a centralized manner in each router. For a description of the LS algorithm, refer to Section 2.2. The next section discusses multipath extensions to the LS algorithm to compute multi-option paths, and the section following shows how the proposed forwarding method is applied.

RED HAT 1012
RED HAT V. CAS

6.3.1 Multipath LS Extensions

This section demonstrates how the proposed forwarding method applies to link state multipath algorithms that calculate suffix matched paths. Since routers in the link state algorithm have access to the entire network topology, any centralized graph algorithm for computing multiple, suffix matched paths can be used. As this section will show, the k ranked paths algorithm [45], and the initial link disjoint k paths algorithm [156] are in this class. For such algorithms, our multipath forwarding algorithm can be directly applied.

As in MPDV, let $\phi^k_{r,d}$ be the identifier that router r assigns its kth path to destination d. The path ID used by router r is of the form $pid = [d, \phi^k_d]$. Forwarding table entries have two additional elements, ϕ_1 and ϕ_2. An entry in router r for the path $(r, s, \ldots, d)$ is a quadruple (d, s, ϕ_1, ϕ_2). ϕ_1 is r's identifier for the path $(r, s, \ldots, d)$ and ϕ_2 is neighbor s's identifier for the corresponding suffix path $(s, \ldots, d)$. Note that $N^r_{d,\phi_1} = s$.

Forwarding entries are installed by the centralized link-state path calculation process. As in MPDV, ϕ_1 is chosen to be unique in r's set of paths identifiers for destination d (i.e. Φ^r_d). To determine ϕ_2 router r calculates the next-hop neighbor router s's paths set for destination d. Since the path sets are assumed to be suffix matched, s's path set to d is guaranteed to include the suffix path $(s, \ldots, d)$. Then, r sets ϕ_2 equal to s's identifier ϕ_1 for that suffix path.

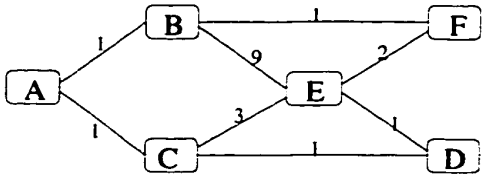
As in MPDV, the functions F and G can now be implemented directly via lookup of the forwarding table. Node r evaluates $F_r([d, \alpha])$ as $[d, \beta]$ via lookup of the forwarding table entry for d and local identifier $\phi_1 = \alpha$, with $\beta = \phi_2$. Similarly, r evaluates $G_r([d, \alpha])$ as $N^r_{d,\alpha}$ via lookup of the same forwarding table entry.

The number of messages exchanged by MPLS routers is exactly the same as in the traditional LS algorithm. The storage overhead per router is $O(NK)$ for the forwarding tables, where N is the number of destinations and K is the number of paths to each destination. However, our multipath forwarding method requires each router to calculate each of their neighbor's K paths as well. Therefore, the suffix matched forwarding method increases the computational complexity of the multipath route calculation used in the routers by a factor

RED HAT 1012
RED HAT V. CAS

m, where m is the number of neighbors.

6.3.2 Example of LS Extension



A's forwarding Table to F

Dst	next-hop	Φ_1	Φ_2
F	B	1	1
F	C	2	2
F	C	3	3

C's forwarding table to F

Dst	next-hop	Φ_1	Φ_2
F	A	1	1
F	D	2	1
F	E	3	1

E's forwarding table to F

Dst	next-hop	Φ_1	Φ_2
F	F	1	1
F	C	2	2
F	D	3	1

Figure 6.4 : Example of the multipath Link State Algorithm.

Consider the network shown in Figure 6.4. The path calculation algorithm calculated three paths from router A to F: (A, B, F), (A, C, D, E, F), (A, C, E, F), with costs 3, 5, 6 and names $\phi^1_{A,F} = 1$, $\phi^2_{A,F} = 2$, $\phi^3_{A,F} = 3$ respectively. Suppose A sends a packet to F using its third best path. Following our algorithm, A tags the packet with the ID [F,3], where F is the destination address and 3 is the ϕ_2 for that forwarding table entry. A then forwards the packet to C, the next-hop node for this path. When C receives this packet, it searches its forwarding table to find an entry with the destination address and $\phi_1 = 3$. C's third forwarding entry matches this path ID. C then replaces the packet's old path ID with [F,1] and forwards the message to E, the next-hop element of the matching entry. That is, $\mathrm{F}_C([F, 3]) = [F, 1]$ and $\mathrm{G}_C([F, 3]) = E$. Similarly, E matches the ID [F,1] and forwards the packet with the ID [F,1] to F.

6.3.3 Suffix Matched Multipath Routing Algorithms

The previous section presented an efficient forwarding method for link state based multipath calculation algorithms that calculate suffix matching path sets. Several existing mul-

RED HAT 1012
RED HAT V. CAS

tipath algorithms generate suffix matched paths. Here we show that the k ranked path algorithm [45, 58, 141, 165] produces suffix matched paths.

Proposition: The unconstrained k shortest (ranked) simple paths algorithm [58] produces suffix matched path sets.

Proof: The unconstrained k shortest path algorithm uses the following invariant for computing multiple paths between all pairs of nodes in the network. A path of rank k between node x_1 and x_n is composed of subpaths all of whose ranks are $\leq k$. Since each node maintains k paths between itself and every other node, all subpaths of a k ranked path between x_1 and x_n are guaranteed to exist in the multipath set computed by the algorithm. QED.

The k initial link disjoint paths algorithms proposed by Topkis [156] also produces suffix matched path sets. In addition, we have already shown in Section 6.2 that any DV based multipath algorithm that satisfies two straightforward conditions produces suffix matched paths.

6.3.4 Non-Suffix Matching Paths in LS

The previous section shows that several important multipath calculation algorithms produce suffix matched paths and therefore can directly use the suffix matched forwarding method. However, there are many other path calculation algorithms, including algorithms that maximize flow between nodes, that are not guaranteed to produce suffix matched paths. Unfortunately, for these algorithms, the suffix matched forwarding presented cannot be directly applied.

To use the suffix matched forwarding method for algorithms that do not calculated suffix matched paths, one needs to convert the non-suffix matched path set into a suffix matched one. We call the conversion of a path set P into a suffixed match set as "forming a *suffix matched closure* over path set P". In general, suffix matched closures can be formed by 1) deleting paths that are not suffix matched or 2) adding paths which are suffixes of non-suffix matched paths. The solutions proposed in this section uses the latter approach. The former approach is not mentioned here because deleting paths reduces the benefits of

RED HAT 1012
RED HAT V. CAS

multipath routing and does not increase the efficiency of forming suffix matched closures. The following lists possible methods to forward packets along non-suffix matched paths.

1. Lazy Compute: this method forms suffix matched closures by dynamically computing non-suffix matched paths. The algorithm works as follows: upon receiving a packet with a path ID that is not suffix matched (this is easily detected by a failure in the path ID match), a router dynamically recompute the path and forward the packet to the right next-hop. The newly computed path ID is then inserted into the forwarding table so that subsequent packet with the same ID do not need to be computed.

   The disadvantage of this method is the potential slow down in packet forwarding time for these paths because some packets require dynamic path computation. However, this disadvantage is reduced because the lazy computation occurs once per path, not once per packet on that path. The computational complexity of the lazy compute method is $O(W + pL)$, where $O(W)$ is the time needed to compute all pairs K paths, p is the number of non-suffix matched paths, L is the average path length (notice that every routers on a non-suffix matched path need to dynamically compute the path). The forwarding table storage costs of this method is $O(KN + pL)$.

2. Explicit Routing: this method is similar to lazy compute except that computation is exchanged with communication and that path IDs are of their original form (i.e. $[Dst, i]$). In this method, when a router receives a path ID that it cannot match (i.e. it is a non-suffix matched path), the router source routes the packet (notice that the router has already computed the path during its path computation phase). As the packet is source routed through the network, it establishes labels along the specified non-suffix matching path. Once the labels are established, subsequent packets that are destined on the same path are forwarded as if the path was suffix matched.

   The advantage of this solution is that it requires no extra router computation. The disadvantage is that packet header sizes need to accommodate source routes, which may complicate packet forwarding procedures. However, like lazy compute, this

RED HAT 1012
RED HAT V. CAS

disadvantage is reduced by the optimization that packets on non-suffix matched paths are source routed only once per path.

3. Common Naming: the idea behind this method is for each router to first calculate a base suffix matched path set that establishes common path ID naming, and a desire set of paths are then chosen from the paths in this base set.

   This method works as follows: first, every router computes M shortest loop-free paths to all other nodes, $M > K$ [58, 103, 141, 165]. We call this set CM (the common naming set). This base set is guaranteed to be suffix matching. Each router then picks K paths from CM – these K paths are the ones computed by a non-suffix matched path calculation algorithm. Let the ith path in K, denoted as K_i, match path j^{th} path in CM, denoted as CM_j. Now when a router forwards a packet on its path K_i, it will use the path ID CM_j. The next-hop router is guarantee to match CM_j because the set CM is suffix matched.

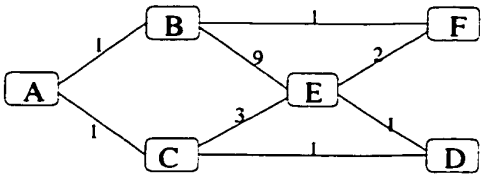
   The advantage of this method is that no extra routing messages are required, and nominal additional router computation are needed to establish the common naming. Moreover, unlike the lazy compute method, the path IDs can be computed *a priori*. The main disadvantage of this method is that routers need M forwarding table entries for each destination instead of K. That is, the per router space complexity of this method is $O(MN)$ instead of $O(KN)$.

The choice of methods to account for non-suffix matched paths depends on the network environment and the efficiency of different path calculation algorithms. The three different methods make different tradeoffs between router computation, routing messages, router storage, and packet overhead. Lazy compute and explicit routing are best suited for large networks where the percentage of non-suffix matching paths are low. Common naming imposes a constant CPU overhead (to calculate M shortest paths) and its efficiency is independent of the percentage of non-suffix matching paths; however, the method requires more forwarding table space.

RED HAT 1012
RED HAT V. CAS

6.4 A Multipath Forwarding Example

This section demonstrates the usage of the suffix matched forwarding method in a general multi-service multi-option network. Recall that a path identifier ψ is a triple (Dst, ψ_1, ψ_2) consisting of the destination address, a service identifier ψ_1, and an identifier specifying a path within that service ψ_2. Section 6.1.1 showed that encoding ψ_1 requires a simple service identifier to uniquely identify different path services, and the previous two sections showed that using suffix matched forwarding, ψ_2 can be encoded as a fixed-sized integer. Notice that both ψ_1 and ψ_2 has a fixed-sized per packet overhead and requires router memory overhead proportional to the number of services and paths within each service. Therefore the path identifier $[Dst, \psi_1, \psi_2]$ is also fixed-sized and requires router storage overhead proportional to the number of paths.



A's forwarding Table to F

P_serv	next-hop	Φ_1	Φ_2
1	B	1	1
1	C	2	2
2	B	1	1

C's forwarding table to F

P_serv	next-hop	Φ_1	Φ_2
1	A	1	1
1	D	2	1
2	E	1	1

E's forwarding table to F

P_serv	next-hop	Φ_1	Φ_2
1	F	1	1
1	C	2	2
2	F	1	1

Figure 6.5 : Example forwarding tables in a network with multi-service and multi-option paths.

Figure 6.5 shows a six node network with link cost shown above the network edges. In this example, the network provides two path services, numbered 1 and 2. The multi-service and multi-option paths can be seen in the router forwarding tables given in Figure 6.5. The forwarding tables for A, C, and E show that the first path service (denoted by P_serv) contains two paths (denoted by the ϕ_1 column), and the second service has one path. The tables show only the forwarding entries to node F.

In this example, assume A wishes to send a packet to F on the first service's second

RED HAT 1012
RED HAT V. CAS

path, which is the path (A, C, D, E, F). To do this, A tags the packet with the path ID $[F, 1, 2]$ and send it to C according to its forwarding table. On receiving this packet, C matches the destination address, path service number, and ϕ_1 to find that the next-hop for this path is D. According to the suffix matched forwarding method, C sends the packet to D with the path ID $[F, 1, 1]$. After D processes this packet, it sends the packet to E with the ID $[F, 1, 1]$. E then forwards the packet to F, thereby successfully forwarding A's packet on path (A, C, D, E, F).

Forwarding a packet on A's second service path is accomplished in the same way. The second service path from A to F is (A, B, E, F). Although the ϕ_1 and ϕ_2 entries are not necessary for single-option paths, it is shown in the Figure 6.5 to demonstrate that single-option forwarding is a special case of multi-option suffix matched forwarding. In this example, A sends a packet to B with the path ID $[F, 2, 1]$. B receives this packet and forwards to E with the ID $[F, 2, 1]$. Upon receiving this packet, E looks up its forwarding table for destination F and path service 2. According to the result of this lookup, E forwards this packet to node F with path ID $[F, 2, 1]$.

6.5 Multipath Forwarding Summary

One of the main challenges of implementing a multipath network is solving the path forwarding problem, defined as delivering packets on their intended paths. Because each router individually decides where to forward packets, ensuring that a packet travels its chosen path requires an agreement among all routers in the network. This path agreement is encoded in the form of a path ID. This chapter develops a packet encoding and forwarding method for multipath networks that uses a fixed-sized path ID and requires routing storage proportional to the number of paths calculated.

The forwarding problem is solved differently depending on whether packets are forwarded on multi-service or multi-option paths. This chapter addresses the two cases individually and combines the solution to solve the general multi-service, multi-option forwarding problem. Because paths in different services can be consistently distinguished,

RED HAT 1012
RED HAT V. CAS

tagging packets with a path service identifier sufficiently distinguishes packets traveling on different services. However, specifying multi-option paths is not so straightforward because multi-option path numbers do not necessarily denote consistent paths on different routers.

To address this issue, chapter presents a novel multi-option forwarding method that uses small, fixed-length path identifiers and requires router storage proportional to the number of paths calculated. When compared to the general method for multi-option forwarding with fixed-length identifiers, the proposed method reduces the router state space needed for packet forwarding by a factor of L, where L is the average path length in the network. This chapter defines the *suffix matched* property for multipath sets and shows that many important multipath calculation algorithms produce path sets with this property. The suffix matched forwarding method presented in this chapter exploits this property.

This chapter also applies the method to two well known classes of routing algorithms—the distributed DV and the decentralized LS algorithms. For multipath extensions of DV, the storage and message overhead required is proportional to the number of extra paths calculated. In the LS multipath extension, no additional messages are needed. The additional Link State computation and storage cost for suffix matched paths are also proportional to the extra number of paths calculated. For paths calculated by non-suffix matched calculation algorithms, efficient methods to forward packets on these paths are also presented in this chapter.

RED HAT 1012
RED HAT V. CAS

Chapter 7

Multipath Transport Protocol

The three elements necessary to make multipath networks viable are 1) appropriate paths calculated between nodes, 2) efficient packet forwarding on calculated paths, and 3) effective end-host usage of multiple paths. Chapter 5 addresses the first issue by developing path calculation algorithms that provide quality paths between nodes. Chapter 6 addresses the second issue with an efficient multipath forwarding method. This chapter addresses the third issue by developing an end-host protocol that effectively uses multiple paths to increase end-to-end throughput.

A node's method of transmitting data has a large impact on network and end-to-end performance. In the current Internet, TCP is the predominant end-host transport protocol that manages the transmission of data [124]. Studies have shown that the transmission mechanisms of TCP have dramatic performance impacts on both end-to-end performance and the performance of other connections [18,27,83,119,120,123]. In a multipath scenario, the way in which nodes transmit data has an even greater performance impact because nodes not only have to decide how much data to send on each path (as in the single path environment), they also have to decide how to distribute the data over multiple paths. As Section 4.2.4 showed, the way in which multiple paths are used can result in significant performance enhancement or degradation.

This chapter develops the MPTCP transport protocol that effectively manages multiple paths to increase end-to-end throughput. MPTCP demonstrates that applications can immediately reap the benefits of multipath networks without any application modifications.

The remainder of this chapter is organized as follows. The next section addresses the issue of usage layer and presents arguments in favor of developing MPTCP. Section 7.2

RED HAT 1012
RED HAT V. CAS

describes how multiple paths should be used in order to increase throughput. Section 7.3 describes the single path TCP protocol. Then in Section 7.4, the MPTCP algorithm is presented. Simulation results that show the effectiveness of MPTCP are provided in Section 7.5. The conclusion of this chapter is that MPTCP is able to effectively use multiple paths to improve end-to-end and network-wide throughput.

7.1 Usage Layer

The multipath usage layer is the protocol layer responsible for distributing data over multiple paths. The choice of which layer should manage multiple paths depends on performance issues as well as software engineering concerns such as system compatibility, maintainability, and clean layering abstractions. There are three possible usage layers in the current TCP/IP protocol stack: network, transport, and user/application. A description of the different usage layers and our motivation for choosing the transport layer are given below.

The lowest usage layer is the *network usage layer*. Implementing multipath routing at this level means the decision of which packet should travel which path is made in the network protocol layer. In the current Internet, this implies that the IP layer must be modified to multiplex data among multiple paths. The advantage of this usage layer is that protocol layers above the network layer do not need modifications in order to use the multiple paths. However, this usage layer has two severe limitations. First, because the network layer is unaware of the demands of the user/application, the IP layer does not have the necessary information to properly distribute data onto paths that best improves application performance. Second, because higher level protocols are unaware of the multipath capabilities of the network layer, events such as out of order delivery and large delay variances may adversely affect the performance of these protocols [83].

The second possible usage layer is the *transport usage layer*. This usage layer shifts the responsibility of multipath management one level up in the protocol stack to the transport layer. The principal disadvantage of this usage layer is the required modifications to the

RED HAT 1012
RED HAT V. CAS

transport and network protocols. Network protocols need to be modified to understand multiple paths to the same destination, and transport protocols need modifications to manage these paths.

The transport usage layer has three principal advantages: 1) the transport layer has the proper information to efficiently use multiple paths because it is already responsible for connection based congestion and flow control. 2) Unlike the network usage layer, the transport protocol performance will not be degraded by side-effects of using multiple paths (e.g. out-of-order deliveries and large delay variances) because the protocol itself knows that it is transmitting on different paths. 3) Since the transport layer is responsible for packet fragmentation and reassembly, it is able to use multiple paths with great flexibility.

The third possible usage layer is the *application usage layer*. This layer offers the greatest flexibility and information to achieve high performance, but requires that application developers have expert knowledge in path management such as congestion and flow control. Although there may be few specialized applications that can use this usage layer, expecting general application developers to have sufficient knowledge to effectively use multiple paths is unrealistic. As an example, one cannot assume that most application developers today can manage data transfer on a single path as good or better than TCP.

After considering the possible usage layers, we chose to develop a multipath protocol in the transport usage layer. The transport layer was chosen because it provides a simple mechanism that allows general users to obtain performance benefits from multipath networks. The protocol developed, MPTCP, maximizes end-to-end throughput. Maximizing throughput was chosen because it is a general service that can be used by many applications and is a standard measurement of network performance.

The purpose of developing the MPTCP protocol is to demonstrate that applications (end-users) can benefit from a multipath network without application changes. We envision that when multipath networks become prevalent, different transport protocols will be developed that cater to different application needs. For example, applications with QoS requirements will use a QoS transport protocol. Similarly, a transport protocol may be

RED HAT 1012
RED HAT V. CAS

developed in a pricing network that minimizes the monetary cost of sending messages. Notice that because this chapter develops a multipath transport protocol, this does not preclude specialized applications from developing their own transfer protocols. In fact, we believe applications such as Web browsers may do exactly this to increase their performance.

7.2 Throughput Optimization

One of the benefits of multipath routing is that end-hosts can aggregate the resources of multiple path to increase performance. In maximizing throughput, the appropriate strategy is to use multiple paths concurrently. However, the amount of data that should be transmitted on different paths depends on the current availability of network resources.

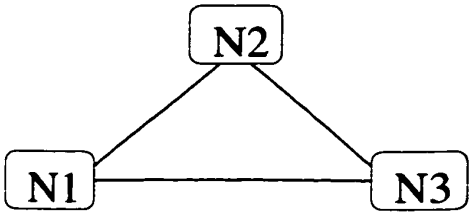


Figure 7.1 : A simple three node network. All links have equal bandwidth and latency.

For the purpose of explanation, consider a simple three node network shown in Figure 7.1 where all links have uniform bandwidth and delay and that the routing algorithm calculates two link disjoint paths between every node pair.

In this network, assume that N1 wishes to maximize its data transmission to N3. If N1 is the only node using the network, then N1 can obtain twice the throughput if it uses two paths (paths (N1, N3) and (N1, N2, N3)) than if N1 only uses the shortest path (N1,N3). Because N1 is the only node using the two paths, there is no contention for network link bandwidth, allowing N1 to fully utilize both paths.

Now consider the scenario where both N1 and N2 are maximizing their throughput to N3. Assuming that both nodes use their two paths to N3 and that the nodes share contended link bandwidth equally, then each node's individual throughput will be the same as if the

RED HAT 1012
RED HAT V. CAS

nodes only used one path. The reason is that the link (N2, N3) is being used by both connections, as is link (N1, N3). Because of this contention, each node only obtains half the link bandwidth from each path.

When three nodes transmitting at the same time, one can see that if all three nodes use their two paths equally to maximize their throughput, then each node will only get 1/3 of the bandwidth from each path. Given that each node uses two paths, this means each node obtains 2/3 the throughput of a single path. Notice that if all three nodes simply use the shortest path, their effective throughput will actually be higher. The consequence of naively using multiple paths is confirmed both in Figure 4.3 in Chapter 4 and in experiments presented later in this chapter.

This simple example shows that optimizing performance is a balance between conserving network resources (e.g. using only the shortest path) and employing additional resources (e.g. using multiple paths) to surpass single path performance. First, at low levels of network utilization, the optimization point favors using multiple paths because there are plenty of underutilized network resources; thus using them allows increased throughput and does not noticeably affect other connections. However, as network utilization increases and the amount of underutilized resources decreases, the performance optimization point shifts. After this point, a strategy that conserves resources achieves better performance because the amount of additional resources used to deliver packets on non-shortest paths causes enough contention to noticeably degrade performance of other connections, while at the same time, offering little throughput gains for the transmitting node.

Thus one of the goals in designing a throughput based multipath protocol is to ensure that the protocol can adapt to this shift in optimization point. The MPTCP protocol developed in this chapter has this adaptive ability. The description of the protocol is given in succeeding sections.

RED HAT 1012
RED HAT V. CAS

7.3 TCP

Before presenting the MPTCP protocol, this section first describes the single path transport protocol, TCP, on which MPTCP is based. Transmission Control Protocol (TCP) is an end-to-end protocol that provides a reliable bit stream service between application programs [83]. That is, it ensures that the destination application receives the same data stream sent by the sending application. Furthermore, TCP aims to fully utilize the available bandwidth of the path it uses. In other words, TCP aims to maximize a connection's throughput while avoiding path congestion. Due to its efficiency and application interface, TCP has become the predominant transport protocol in today's Internet [122]. A brief description of TCP's data transfer mechanisms are given below.

In TCP, the TCP sender sends data to a TCP receiver using variable sized packets called segments. For every segment sent by TCP, a sequence number is assigned to the segment. These sequence numbers are used by the receiver to reassemble segments into their original order and to acknowledge the reception of each segment. If the receiver does not acknowledge a segment, the TCP sender will retransmit the unacknowledged segment. This acknowledgment and retransmission scheme ensures that all data transmitted are eventually received.

In addition to reliably transmitting data, TCP also provides flow control and congestion control. Flow control ensures that the sender only sends as much data as the receiver can buffer, and congestion control ensures a TCP connection never overloads a path with data. That is, congestion control ensures that the path a TCP connection is using never remains in a state of congestion. Congestion and flow control is achieved by the use of a sliding window called the congestion window. The size of the window varies to indicate the amount of unacknowledged data a sender is allowed to send without overloading the path or the receiver.

Congestion control in TCP consists of three algorithms: slow start, congestion avoidance, and fast retransmit and fast recovery. The slow start algorithm is active when the congestion window is below a certain threshold called *ssthresh*. The sender in slow start

RED HAT 1012
RED HAT V. CAS

increases its congestion window by one segment each time an acknowledgment (ACK) is received. Thus in slow start, every ACK causes the sender to send two new segments: the first segment is sent to fill the congestion window made available by the ACK and the second segment by slow start's increase of the congestion window. In slow start, the size of the congestion window is doubled each round trip time (RTT). The motivation for the slow start algorithm is to quickly probe the amount of available path bandwidth.

The congestion avoidance algorithm is active when the congestion window increases beyond ssthresh. In congestion avoidance, the sender increases its congestion window by one segment every RTT. In this mode, the sender sends one new segment for every acknowledged segment, and after n segments have been received where n is the current congestion window size, the sender sends one additional segment. The goal of congestion avoidance is to slowly probe the path for additional bandwidth.

The third TCP congestion control algorithm is the fast retransmit and fast recovery mechanisms which deal with TCP's response to lost segments. In TCP, a lost segments is detected when the sender does not receive an ACK for the segment. A timeout mechanism is used to trigger the sender to send unacknowledged segments; however, the timeout mechanism typically takes a long time and thus reduces TCP performance. The fast retransmit mechanism speeds this process by retransmitting a "lost" segment after receiving three *duplicate ACKs*. A TCP receiver sends a duplicate ACK whenever it receives a segment that is not the next segment in the sequence number. Using fast retransmit, the TCP sender assumes that receiving three duplicate ACKs indicates that the oldest unacknowledged segment never reached the receiver.

The fast recovery mechanism further optimizes fast retransmit by adjusting the congestion window such that after a fast retransmit, the TCP sender is allowed to send half a congestion window's worth of new data. This optimization avoids stalling the sender while waiting for the lost segment's ACK. When the lost segment's ACK is received , the fast recovery algorithm terminates and reduces the TCP congestion window by *half*. After the termination of fast retransmit and fast recovery, the TCP connection continues transmission

RED HAT 1012
RED HAT V. CAS

in congestion avoidance mode.

For more information about the TCP protocol, refer to [12,27,78,83,84,119,128,168].

7.4 MPTCP

This section describes the implementation of a reliable bit stream multipath transport protocol. The protocol, called MPTCP, aims to maximizes end-to-end throughput. MPTCP requirements are that it

1. Performs congestion and flow control among multiple paths
2. Provides a reliable bit stream service (same as single path TCP)

The first requirement states that MPTCP must be sensitive to network congestion and not overrun the receiver. Congestion control allows high effective throughput even at high levels of network utilization. The second requirement states that an MPTCP receiver must receive the same bit stream sent by the MPTCP sender. In using multiple path, this implies that MPTCP need to fragment a sender's data stream, send data on multiple paths, and reconstruct the original data stream at the receiver.

The approach taken to implement MPTCP is to modify the single path TCP to use multiple paths. This approach has two main advantages. First, by extending TCP, one is able to leverage the algorithmic advancements made in TCP; thus the performance improvements made in TCP over the past decade can be directly translated into MPTCP performance. The second advantage is that it allows MPTCP to readily take advantage of future TCP improvements because MPTCP can easily upgrade whenever TCP upgrades.

7.4.1 The MPTCP Algorithm

This section describes the MPTCP algorithm. The base TCP algorithm used in MPTCP is TCP New-Reno [84]. New-Reno was chosen because of its wide use and advanced congestion control mechanisms. These mechanisms were described in the previous section.

RED HAT 1012
RED HAT V. CAS

MPTCP extends TCP very naturally. When a sender opens an MPTCP connection to a destination, MPTCP opens K concurrent and independent TCP connections to the same destination, where K is the number of paths the network provides between the sender and receiver. In MPTCP, the TCP connection establishment procedure is unaltered (e.g. a TCP three-way handshake is performed on each connection). Whenever the sender wishes to send a data stream, it passes it to MPTCP. MPTCP then divides this data stream into MPTCP segments and sequence numbers each segment*. The size of an MPTCP segment is such that it is no larger than the size of the underlying TCP segment size (i.e. TCP's minimum transmission unit) minus the length of MPTCP control information such as MPTCP sequence number and MPTCP segment size. Because the MPTCP segment size does not exceed TCP segment size, this ensures that TCP does not fragment an MPTCP segment in order to send the data, increasing MPTCP's overall efficiency. Notice that MPTCP's different sub-TCP connections may have different segment sizes; therefore the size of an MPTCP segment size varies depending on the TCP connection it is sent on.

When a destination TCP connection receives segments from its TCP sending peer, it reconstructs the received messages in the usual manner. The MPTCP receiver then reads the TCP data stream to recover MPTCP control data. Using the control data, MPTCP receiver then reconstructs the original MPTCP data stream from all its sub-TCP connections. This data stream is then returned to the receiving application. Notice that MPTCP does not need to explicitly acknowledge segments because the underlying TCP connections ensure reliable data delivery.

To make maximum use of each TCP connection, MPTCP sends on each of its TCP connection the number of segments allowed by the connection's congestion and flow control mechanisms. To do this, MPTCP provides the next MPTCP segment to a TCP connection only when the connection is ready to send a new segment of data. This dynamic load balancing allows MPTCP to fully utilize each TCP connection.

---

*Notice there are two levels of sequence numbers, one for MPTCP and the other for TCP. The two sequence numbers are independent.

RED HAT 1012
RED HAT V. CAS

Because MPTCP uses TCP to transmit data, it inherits TCP's congestion and flow control mechanisms. For example, when an MPTCP's sub-TCP connection detects congestion, the sub-connection will decrease its sending rate in the same manner as a normal TCP connection. Thus, MPTCP's congestion control is as good as TCP's congestion control. MPTCP performs flow control in the same way. Figure 7.2 shows an example MPTCP connection.

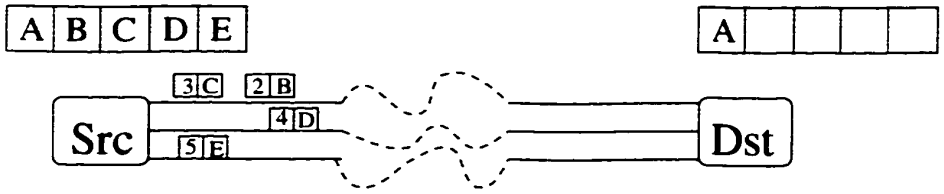


Figure 7.2 : This example shows an MPTCP connection using three paths. The original message is disassembled and sent on the three available paths using TCP. The destination node reassembles the original message using MPTCP sequence numbers.

In Figure 7.2, Src uses MPTCP to send a data stream to Dst. The figure shows that the Src MPTCP opens a TCP connection on each of the three paths provided by the network. The TCP connections are represented by the horizontal lines in the figure. The sender passes a message stream to MPTCP which are divided into five segments, labeled A through E. MPTCP sequence numbers each segment and sends them onto the different TCP connections. The boxes above each TCP connection in Figure 7.2 denote TCP frames being transmitted by that connection. The number in the box denotes the MPTCP sequence number, and the letter denotes the data segment. In this example, the MPTCP receiver has already received segment A.

7.4.2 MPTCP Fairness

In a realistic multipath network, nodes using MPTCP will compete with nodes using other protocols such as single path TCP. This scenario raises the concern that nodes using MPTCP have an unfair advantage over TCP nodes because MPTCP not only uses multiple paths but

RED HAT 1012
RED HAT V. CAS

also competes for bandwidth equally with other TCP connections. This section extends the MPTCP algorithm to address this fairness concern.

Before addressing MPTCP and TCP fairness, it is worth noting that TCP itself does not provide fairness among competing TCP connections [168]. For example, in all current implementations of TCP, if two connections share a link, then the connection with a lower bandwidth-delay product will consume more bandwidth on that link. Furthermore, many researchers such as Keshav and Kalmanek advocate that in order to enforce bandwidth fairness, routers must support more intelligent queuing disciplines than Internet's current FIFO queuing [90, 113, 135]. This implies that MPTCP and TCP fairness is difficult to achieve because the underlying network and TCP implementations do not have fairness mechanisms.

In light of these factors, this section addresses the fairness concern by making MPTCP secondary connections less aggressive in competing for bandwidth. A MPTCP secondary connection is a connection that is not using the first path (primary path) in a path set. Details of our solution are given below.

Congestion Backoff Percentage

To make an MPTCP's sub-TCP connection less aggressive, we alter the TCP's congestion backoff strategy. Recall that in normal TCP, whenever the protocol senses network congestion (e.g. via unacknowledged packets), TCP reduces its congestion window by $1/2$, or a backoff percentage of 50%. To make secondary connections less aggressive, the proposal sets secondary connection's congestion backoff percentage to $X\%$, where $X \leq 50$. The primary MPTCP connection is unaltered (i.e. its congestion backoff percentage is 50%). The congestion backoff percentage also sets the connection's ssthresh to same value as the reduced congestion window size.

Notice that this method should affect a secondary connection's throughput only when congestion is encountered. That is, in the absence of contention from other connections, a secondary connection should still exploits the bandwidth of a path as before. A secondary

RED HAT 1012
RED HAT V. CAS

connection's performance should differ only when competing connections cause link congestion. In this case, secondary MPTCP connections will back off more, thus giving the TCP connections using that link more share of the bandwidth.

Unfortunately in practice, this is not always true: in TCP-new Reno for example, because the protocol probes network bandwidth by causing congestion, secondary connections will back off more even though there may not be any contending connections. On the other hand, with more advanced TCP algorithms that do not probe path bandwidth through causing congestion [27], their TCP secondary connections should behave the same in the absence of congestion.

The impact of congestion backoff percentage on TCP fairness and MPTCP performance is shown in Section 7.5.

7.4.3 Path Information

The previous sections describe the MPTCP protocol; however, they did not describe how MPTCP obtains path information and how MPTCP specifies a packet's path. This section discusses these two issues.

For MPTCP to obtain information about the paths a network provides (both the type of services and the number of paths in each service), it must query its local router. The result of the queries should be a set of path IDs that the router calculates between network nodes. Querying the local router is sufficient because the router knows the type and the number of paths calculated to different destinations. How this information is exchanged and stored between hosts and local routers depend on the details of the actual network. We believe that this information can be efficiently propagated using a protocol similar to ones used in local area networks to distribute information (e.g. NIS, ARP, and DNS). In addition, because this information sharing is purely local, it does not have significant impact on the scalability, cost, or performance of multipath networks. For these reasons, the actual protocols to propagate this path information are not described in this thesis.

In order to specify a path in a multipath network, a packet needs to specify a destination

RED HAT 1012
RED HAT V. CAS

address and a path identifier. The forwarding method developed in Chapter 6 requires that each packet is tagged with a path ID of the form $[Dst, \psi_1, \psi_2]$ where Dst is the destination address, ψ_1 specifies the path service, and ψ_2 specifies the particular path within that service. Our implementation of MPTCP uses this path ID format.

In our simulator, IP headers are augmented with a path ID field. This field is interpreted by routers to determine where to forward a packet; the forwarding procedure is given in Chapter 6. Whenever MPTCP sends a segment, it sets the segment's path ID field, thereby ensuring that the segment travels the specified path.

7.4.4 Limitations

The MPTCP protocol described in this chapter uses multiple, independent TCP connections. That is, the different TCP sub-connections operate independently and are unaware of each other. The advantage of this approach is that the MPTCP protocol makes minimal modification to TCP, which allows MPTCP to easily change and upgrade the underlying TCP protocol. The disadvantage is that the different connections are not able to work together. This section lists some of the limitations of MPTCP as well as some possible improvements.

One limitation of non-cooperative sub-connections is that a sub-TCP connection cannot reduce the retransmission duties of another connection. For example, consider an MPTCP connection that opens three TCP sub-connections and that connection 1 has the highest bandwidth path to the destination. Suppose that during data transmission, this connection's path becomes very congested and subsequently drops many of this connection's packets. In MPTCP, connection 1 is solely responsible for retransmitting all the lost segments; this may be many segments depending on the congestion window size and the severity of the congestion. In an ideal situation, however, MPTCP should be able to transmit connection 1's lost segments on different connections in order to off load the connection's retransmission duties.

Notice that *not* off loading connection 1's unacknowledged segments to other con-

RED HAT 1012
RED HAT V. CAS

nection may not reduce the amount of segments the sender sends to the receiver because MPTCP dynamically load balances between TCP connections. That is, MPTCP will not give new data to TCP connections that are not ready to transmit new data. Despite dynamic load balancing, requiring a connection to retransmit a large number of segments forces the MPTCP receiver to buffer more segments from other connections. This results in higher receiver memory usage, increased MPTCP connection latency, and potentially lower throughput.

Another example where cooperating TCP connections can improve performance is where a connection's forward path is underutilized but the reverse path is congested. In this scenario, even though a TCP connection's forward path can transmit more segments, TCP will not send them because the ACKs are returning slowly or are being lost on the reverse path. Ideally, in this case, different sub-TCP connections should cooperate such that the receiver sends ACKs on the least congested reverse path. This way, the sender can fully exploit the bandwidth of its forward paths. However, this optimization is not possible in MPTCP because its sub-TCP connections are independent,

In short, designing a multipath transport protocol requires a tradeoff between the sophistication of the protocol and its performance. This section indicates that MPTCP can obtain higher performance if its sub-connections cooperate. However, the potential increase in performance comes at the cost of implementing cooperative mechanisms. In this thesis, we make the tradeoff toward simplicity rather than performance. Simulations show that the despite its simplicity, MPTCP is able to effectively exploit available network resources. These simulation results are given next.

7.5 MPTCP Experiments

This section experimentally evaluates MPTCP performance. Because MPTCP increases end-to-end throughput, the evaluation of MPTCP is in terms of effective throughput. In addition, fairness issues between MPTCP connections and other single path TCP connections are also addressed here.

RED HAT 1012
RED HAT V. CAS

MPTCP throughput is evaluated here not only because throughput is a common network performance metric, but also because many applications such as FTP will benefit from increased network throughput. The experiments conducted in this section examine MPTCP performance with varying network topologies and traffic patterns. The results show that MPTCP effectively increases network throughput, even under high network utilization levels.

Another issue this section addresses is the fairness of MPTCP. Since MPTCP uses more paths and competes with other single path TCP connections, it seems that TCP performance would significantly degrade when competing with other MPTCP connections. The experiments in this section evaluate the performance of TCP connections versus MPTCP connections with different congestion backoff percentages. The result of the simulations show that congestion backoff slightly improves the throughput of competing TCP connections.

The experiments in this section are organized as follows. In Section 7.5.1, the performance of MPTCP is simulated on a simple, 3 node network. Section 7.5.2 uses the same network to evaluate MPTCP fairness with respect to TCP connections. Finally, Section 7.5.3 verifies MPTCP performance and fairness in a more realistic network scenario with an Internet-like network topology and traffic pattern. The conclusion of these experiments is that MPTCP effectively utilizes available network resources and that its increased performance has nominal effects on other TCP connections.

7.5.1 Aggregate Throughput

The experiment conducted in this section measures MPTCP throughput. This experiment uses a basic three node network shown in Figure 7.1. The simple network is used to understand basic MPTCP behaviors.

In the network depicted in Figure 7.1, all links are full- duplex and have equal capacity and delay, 1000KB/s (kilobytes per second) and 10us respectively. In addition, each router can buffer up to 50 packets for each of its out-going links. The multipath routing algorithm provides two link disjoint paths between each node, the one- and two-hop paths. Each node

RED HAT 1012
RED HAT V. CAS

has two clients that send a burst of packets to a random client on another node, and the interval between bursts are exponentially distributed. The average burst interval is varied to change the network utilization level. The burst size is 1000 packets and each packet is 1500 bytes long.

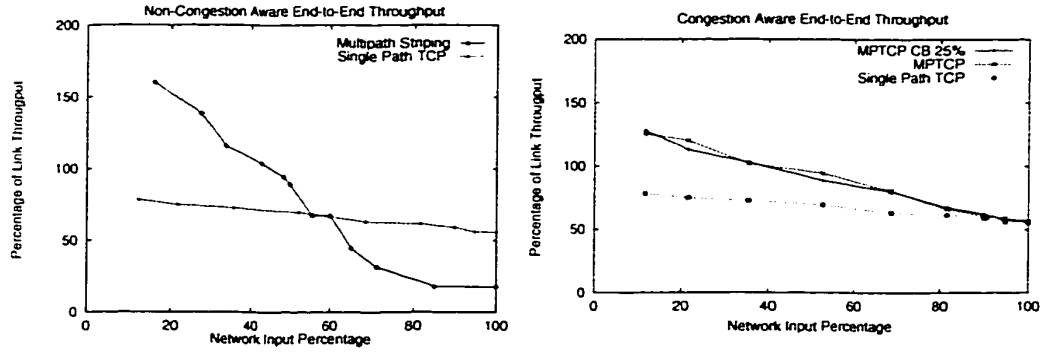


Figure 7.3 : Non-congestion aware multipath protocol versus MPTCP on a triangle network. The left figure shows the throughput percentage of a naive striping multipath protocol versus single path TCP. The right figure shows the performance of MPTCP versus single path TCP.

The graphs in Figure 7.3 features four transport protocols: TCP, naive multipath striping protocol, MPTCP without congestion backoff, and MPTCP with 25% congestion back-off percentage. TCP and the two MPTCP protocols have been described earlier. The multipath striping protocol works as follows: the multipath striping protocol clocks the sending of its data at full link capacity and distributes the data by striping them along the two available paths. That is, given N packets destined for destination D, the protocol sends packet $2i$ on the one-hop path and packet $2i + 1$ on the two-hop path to D, $0 \leq i < N/2$. Although the striping protocol does not perform congestion control, it does acknowledge received packets and retransmit lost ones.

In Figure 7.3, the y-axis represents the average end-to-end client throughput as a percentage of link capacity, and the x-axis denotes the amount of traffic injected into the network, normalized by network capacity. Thus, $x = 100\%$ denotes all nodes (there are 2 traffic generating clients in each node) simultaneously sending packets. The throughput

RED HAT 1012
RED HAT V. CAS

result of each protocol is represented by their respective labeled curve. In each experiment, all clients are using the same protocol to transmit data to each other.

In both graphs, as the frequency of client transmission increases (marked by network input percentage on the x-axis), the average effective performance decreases. This is because as clients send more data, the amount network contention increases, resulting in lower end-to-end average throughput. Notice that although average end-to-end throughput decreases with increase network traffic, the *aggregate* network throughput increases because the network is more fully utilized.

The left graph in Figure 7.3 compares the performance of the naive multipath striping protocol versus TCP. This graph shows that when the network is relatively unused, the striping protocol performs better than its single path counterpart. However, when clients send data at greater than 60% of network capacity, the performance of the striping multipath protocol quickly degenerates. The reason for the degradation is that the striping protocol does not adapt to the shift in performance optimization.

As stated in Section 7.2, in order to maximize throughput in a multipath network, a protocol needs to alter its data transmission strategy in response to network traffic conditions. At low network utilization (low traffic levels), using more network resources (in terms of using more paths) increases effective throughput. However, at high network utilization, the performance optimization point shifts so that conserving resource usage achieves better performance. In the three node topology, the optimization point begins to favor conserving resources when utilization reaches 60%. At this point, a strategy that conserves resources achieves better performance. The inability to adapt to this optimization shift is indicated by the degradation of the striping protocol's performance curve.

The right graph in Figure 7.3 shows that MPTCP adapts to the optimization shift (both with and without congestion backoff). The graph shows the average end-to-end performance when the nodes are using MPTCP versus TCP. Notice that MPTCP performance is consistently better then TCP, even when nodes are transmitting all the time. This demonstrates that MPTCP is effective in shifting its transmission strategy as the performance

RED HAT 1012
RED HAT V. CAS

optimization point shifts. For example, at full network utilization, MPTCP's performance gracefully degrades to the TCP performance, which is the optimal point at full utilization. This ability to gracefully degrade to TCP indicates the effectiveness of MPTCP's congestion control mechanism at exploiting available bandwidth without incurring performance degradation due to network contention.

The right graph in Figure 7.3 also shows the performance difference of non-congestion backoff MPTCP versus MPTCP with 25% congestion backoff percentage. As the performance curves show, the congestion backoff percentage has nominal affects on competing MPTCP performance.

Comparing the two graphs, the naive striping protocol performs better than MPTCP only when the network is less than 20% utilized. The reason is that the MPTCP congestion control mechanisms are conservative and thus take longer to fully utilize available path bandwidth. However, the same conservative mechanisms allow MPTCP to significantly outperform the naive protocol at high network utilization levels.

The result of this experiment is that MPTCP is able to effectively use multiple paths to increase throughput, and that even during high levels of network contention, MPTCP outperforms its single path counterpart.

7.5.2 MPTCP Congestion Backoff Percentage

The previous experiment shows that MPTCP outperforms TCP when all clients are using the same protocol. As mentioned earlier, in a realistic multipath network, nodes using MPTCP will be competing with nodes using TCP. This section examines the impact of congestion backoff percentage on MPTCP performance and the performance of competing TCP connections.

The impact of MPTCP congestion backoff percentages is shown in Figure 7.4. The curves labeled SP-x denote the throughput obtained by a TCP connection while competing with x MPTCP secondary connections (x ranges from 1 to 4). The simulations were conducted on the three node network in Figure 7.1. In this experiment, N1 establishes a

RED HAT 1012
RED HAT V. CAS

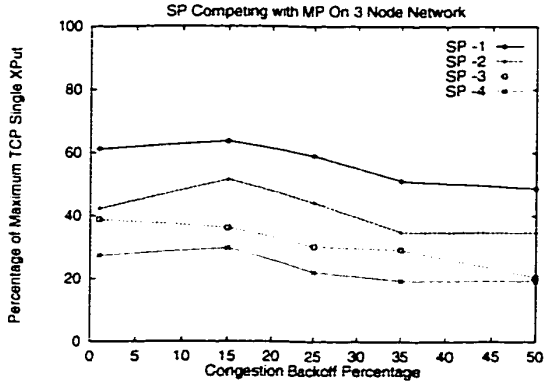


Figure 7.4 : Congestion backoff percentage on a three node network.

TCP connection and sends data to N3 while x MPTCP connections from N2 simultaneously transfer data to N3. Thus, N2's the secondary MPTCP connections compete with N1's TCP connection on link (N1, N3).

In Figure 7.4, the y-axis denotes the TCP throughput as a percentage of TCP's throughput without any competing connections. The x-axis indicates the congestion backoff percentage of the MPTCP secondary connection(s). The throughput of the secondary connections is not shown in the figure.

As the graph shows, TCP is able to obtain higher throughput when secondary MPTCP connections back-off more. However, the performance gain is not very much. For example, consider the SP-1 curve which shows a single TCP connection competing with a single MPTCP secondary connection (the top curve in Figure 7.4), at congestion backoff percentage of 25%, the TCP connection obtained 60% of the available TCP bandwidth. This is only 10% more than competing with a secondary MPTCP connection at 50% congestion backoff percentage.

Contrary to the experimental results, one expects that TCP performance would be more or less inversely proportional to the MPTCP congestion backoff percentages. Detailed analysis of TCP traces shows the cause of the observed behavior. As the congestion backoff percentage decreases, MPTCP secondary connections retract their congestion windows more whenever they sense congestion. This has the temporary effect of giving the TCP

RED HAT 1012
RED HAT V. CAS

connection more of the network bandwidth. However, as the MPTCP/TCP connections resume and increase their sending rates, the network eventually drops packets. When packets are dropped, the majority of dropped packets will be from the TCP connection because this connection is sending the majority of packets on the congested link. These packet drops reduce the TCP congestion window, decreasing its throughput.

Because the probability that a connection's packets are dropped on a congested link is proportional to the number of packets the connection sends on that link, the performance gains TCP obtains from less aggressive secondary MPTCP connections are diminished. Therefore, TCP performance does not significantly exceed the performance of MPTCP secondary connections.

In spite of this phenomenon, decreasing congestion backoff percentage does give TCP connections slightly more share of the network bandwidth. This motivated us to set the MPTCP congestion backoff percentage at 25%.

7.5.3 Foreground and Background Traffic

The previous two sections evaluate the performance and fairness of MPTCP on a three node network with simple traffic patterns. The purpose of this section is to determine whether the MPTCP benefits observed in the three node network can be translated in a larger, more realistic network and traffic model. In addition, fairness is also evaluated in this section by measuring the effect of MPTCP on TCP performance. Specifically, whether MPTCP performance improvements come at the expense of other TCP connections.

To better reflect the Internet topology, this experiment uses a cluster network topology. The simulated network is composed of 195 links and 10 cluster networks, with 10 nodes per cluster. Refer to Section 9.1 on the construction of cluster networks. Every node in the network can send and receive data, and the capacity removal algorithm is used to compute paths between nodes (Chapter 6). For a more realistic traffic pattern, a combination of foreground and background traffic is used. A foreground traffic node transmits a large stream of packets (5,000 packets) to a receiving node, which is randomly chosen with

RED HAT 1012
RED HAT V. CAS

the restriction that it must be in a different cluster than the sending node. A background traffic node sends a stream of packets ranging from 100 – 2,000 packets to another network node chosen at random without restrictions. After transmission, background nodes wait an exponentially distributed amount of time (a Pareto distribution with average of 20 seconds) and then proceed to transfer data to another destination node picked at random. Three of the 100 nodes are randomly chosen to transmit foreground traffic. The remaining 97 nodes generate background traffic. All nodes transmit using either MPTCP or TCP. The results of the simulations are collected when all foreground nodes finish transferring.

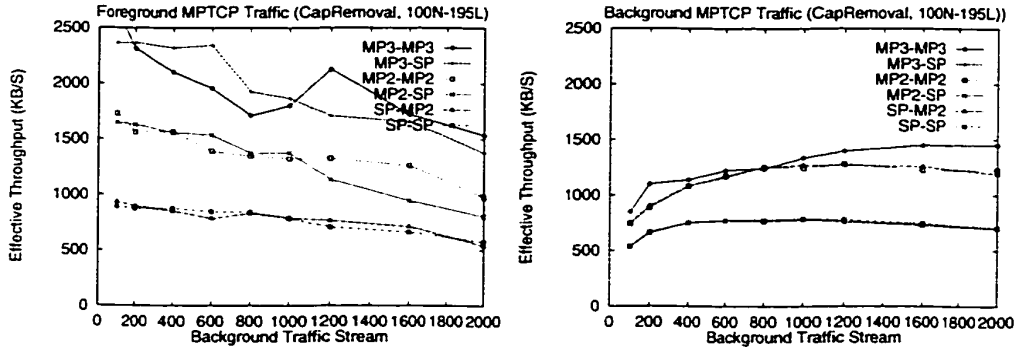


Figure 7.5 : The foreground and background performance of MPTCP using the capacity removal algorithm. The left graph shows the average foreground throughput and the right graph shows the average background throughput.

The two graphs in Figure 7.5 show the MPTCP and TCP foreground and background traffic throughput. In both graphs, the x-axis shows the burst sizes of the background traffic, which range from 100 to 2,000 packets, and the y-axis shows effective throughput in KB/s. Each curve is labeled to represent the transport protocol used by foreground and background nodes. The first label denotes the foreground transmission style, and the second denotes the background transmission style. SP stands for using TCP, and MP2 and MP3 stand for using MPTCP with 2 and 3 paths respectively.

For example, the curve with label MP3-SP denotes that the experiment was conducted where the foreground nodes transmitted using 3-path MPTCP, and the background nodes used TCP. The analysis of Figure 7.5 is divided into MPTCP performance and MPTCP's

RED HAT 1012
RED HAT V. CAS

effect on TCP connections. These analyses are given below.

MPTCP Performance

The left graph in Figure 7.5 shows the foreground performance curves. The lowest two curves show the foreground throughput of TCP connections. The two curves only differ in their background transmission styles. Similarly, the middle two curves show 2-path MPTCP performance, and the two highest curves denote the 3-path MPTCP foreground performance.

Notice that as network utilization increases (marked by increasing background traffic bursts), both 2- and 3-path MPTCP gracefully degrades its performance to reflect shift in the performance optimization point. As in the three node network, this shows that MPTCP is able to effectively increase performance even at high network utilization levels. In addition, this graphs confirms two important points: 1) the capacity removal algorithm effectively offers throughput-service paths, and 2) MPTCP is able to increase throughput given additional network resources.

The right graph in Figure 7.5 shows the background throughput performance. The curves are labeled in the same manner as foreground performance graph. The bottom 3 coinciding curves denote the background TCP performance. The difference in these curves is the foreground transmission style. The middle 2 curves (SP-MP2 and MP2-MP2) show the background performance of 2-path MPTCP. These two curves are just below the MP3-MP3 curve. Notice at low background burst sizes, the background traffic performance does not achieve the same level of throughput as the foreground traffic. The reason is that the small burst sizes were not large enough for the TCP algorithms to achieve steady state. Thus, the background performance differences are not as evident as in the foreground performance curves. Despite the smaller performance distinction, the higher performance achieved by 2-path and 3-path background MPTCP connections again demonstrates the protocol's ability to increase throughput given more network resources.

In summary, the performance result in this subsection confirms that the MPTCP's

RED HAT 1012
RED HAT V. CAS

throughput on the three node network translates to a larger and a more realistic network. In addition, MPTCP's ability to increase network performance when more paths are calculated demonstrates that the capacity removal algorithm successfully provides high throughput paths.

MPTCP and TCP connections

The second issue this experiment reveals is the interaction between MPTCP and TCP connections. Notice that for both the foreground and background graphs, the throughput curves are grouped by transmission modes (i.e. SP, MP2, or MP3). These tight groupings show that the performance obtained by foreground transmission mode is largely independent of background transmission mode, and vice versa. This shows that the MPTCP performance increase has nominal impact on the performance of TCP connections.

A good example is the SP-MP2 and SP-SP foreground performance curves. Here, the two foreground TCP's performance curves are almost identical while SP-MP2 background performance is consistently better than SP-SP background performance. This shows that the background MP2 and SP connections has roughly the same effect on the TCP foreground connections. Similarly, the tight grouping of background performance curves between SP-SP, MP2-SP, and MP3-SP show that the 2- and 3-path MPTCP foreground connections do not noticeably affect the performance of background TCP connections.

However, this result seems counter-intuitive. Intuitively, given that MPTCP uses more network resources to increase its throughput, it seems that this performance increase should come at the expense of the performances of other connections. The reason this intuition is flawed is that it overlooks the fact that network traffic is bursty; therefore even under heavy network usage, there are still underutilized network resources in the network. So if MPTCP mainly uses these resources to increase its performance, it will not significantly degrade the performance of other connections. The tight performance groupings in the simulation results show that most of the performance gains obtained by MPTCP are due to using otherwise underutilized resources.

RED HAT 1012
RED HAT V. CAS

The conclusion of this experiment is that 1) MPTCP successfully increases end-to-end performance and 2) that the majority of performance increase obtained by MPTCP does not come at the expense of other TCP connections.

7.5.4 Network Resources and Throughput

The previous experiments show that using capacity removal paths, MPTCP can increase network performance over TCP; however, the experiments did not show the dependency between the physical network connectivity and multipath routing's performance potential. This dependency is important because it indicates how much a given network can reap the performance benefits offered by multipath routing. The purpose of this section is to determine this dependency by evaluating the relationship between network connectivity and MPTCP performance. To this end, the experiments in this section vary two parameters: 1) the network connectivity and 2) the number of paths calculated between nodes.

The experiment conducted in this section uses a 20 node network with the number of links ranging from 40 to 120 (network connectivity from 2 to 6). The network topology is randomly generated and all links have equal bandwidth and latency. The capacity removal algorithm is used to compute paths between nodes. In the simulations, each node in the network randomly selects another node and sends a burst 5,000 packets. The inter-burst time is randomly and exponentially distributed using Pareto distribution with a 2 second average. This experiment used a large burst size to obtain steady-state throughput and small inter-burst times to keep the overall network traffic high. Figure 7.6 shows the simulation results.

The x-axis in Figure 7.6 denotes network connectivity, and the y-axis denotes throughput of MPTCP and TCP connections normalized by link capacity. Each curve shows the average throughput of all nodes using the transmission style denoted by the curve's label.

This figure shows three noteworthy results. First, 3-path MPTCP achieves the highest performance for all network connectivities, followed by 2-path MPTCP and TCP. This confirms the performance MPTCP benefits observed in the previous experiments.

RED HAT 1012
RED HAT V. CAS

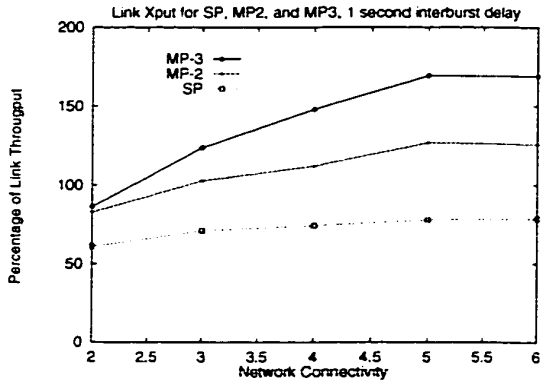


Figure 7.6 : MPTCP throughput on a 20 node network with varying network connectivity.

Second, the performance difference of using multiple paths increases as network connectivity increases. This correlation between performance increase and connectivity shows that the more connected a network, the more end-to-end path resources are available (as calculated by the path calculation algorithm), resulting in higher MPTCP throughput. As a consequence, this implies that the benefits of multipath routing are less in sparse topologies and more in dense topologies.

The third observation is that single path routing severely underutilizes the resources of a well connected network. Notice that the performance of TCP increased only slightly as connectivity increased from 2 to 6. In contrast, 2-path and 3-path MPTCP effectively utilize the available connectivity to increase throughput. Furthermore, the graph shows that 3-path MPTCP obtains higher performance gains than 2-path MPTCP.

These results show that the combination of the capacity removal path calculation algorithm and the MPTCP protocol effectively utilizes available physical network resources to increase end-to-end throughput.

7.6 MPTCP Summary

This chapter develops a transport protocol MPTCP that increases end-to-end throughput using multiple paths. MPTCP provides a reliable bit-stream service and operates by estab-

RED HAT 1012
RED HAT V. CAS

lishing multiple independent single path TCP connections. Moreover, the MPTCP sender dynamically load balances the different TCP connections by sending data on each connection according to the connection's sending rate. The MPTCP receiver then reassembles the data from its receiving TCP connections.

Extensive simulations were conducted in this chapter that demonstrate the effectiveness of MPTCP. The simulations show that MPTCP is able to effectively increase end-to-end throughput in large, Internet-like cluster networks and under both light and heavy network utilization levels. In addition, the experiments demonstrate that the performance improvements achieved by MPTCP do not necessarily come at the expense of other TCP connections, but rather, most of the performance improvements are obtained by using otherwise underutilized network resources. The conclusion of this chapter is that MPTCP effectively utilizes multiple paths to increase end-to-end throughput.

RED HAT 1012
RED HAT V. CAS

Chapter 8

Multipath Routing Algorithms

The previous three chapters presented components necessary to make multipath routing viable: algorithms that compute multiple paths, an efficient multipath forwarding method, and a multipath transport protocol that maximizes end-to-end throughput. This chapter implements two multipath routing algorithms using the developed path calculation and path forwarding components. The two multipath routing algorithms, MPDV and MPLS, based on the Distance Vector and the Link State routing algorithms respectively.

Both routing algorithms developed here use the suffix matched forwarding method and calculate capacity removal paths. Suffix matched forwarding was chosen for its efficiency, and the capacity removal paths are calculated because this path calculation algorithm highlights some key differences between DV and LS path computation styles. Moreover, since the next chapter evaluates the throughput offered by a multipath network, the capacity removal routing algorithms described in this chapter can thus be directly applied to next chapter's experiments.

The description of MPDV and MPLS are given below. Section 8.1 presents MPDV, and Section 8.2 the MPLS algorithm.

8.1 The MPDV Capacity Removal Algorithm

The Distance Vector (DV) routing algorithm is widely deployed in today's networks. Routing protocols that use this algorithm include RIP [7], BGP [133], and EPG. For a basic description of the algorithm, refer to Section 2.1. This section lists the extensions needed to enable a multipath DV algorithm to calculate capacity removal paths.

The description of MPDV is divided into two parts: the additional data structures

RED HAT 1012
RED HAT V. CAS

needed to implement capacity removal MPDV and the actual path calculation algorithm.

8.1.1 Data Structure Costs

This subsection describes the additional data structures needed to convert single path DV (SPDV) to calculate capacity removal paths and to use the suffix matched forwarding method. Data structure changes are needed in MPDV forwarding tables and in MPDV Distance Vector packets (MPDVP's). MPDVP's are multipath versions of single path DVP's, and are exchanged by MPDV routers to compute paths between nodes.

MPDV Forwarding Tables

In the basic SPDV algorithm, each forwarding table entry consists of three elements: the destination address, the next-hop neighbor on the known least-cost path to the destination, and the cost of that path. Extending DV to use the suffix matched forwarding method requires adding two local path IDs to each forwarding table entry, ϕ_1 and ϕ_2 (refer to Section 6.2.2 for details).

However, adding ϕ_1 and ϕ_2 does not provide enough information for MPDV to calculate capacity removal paths. To calculate these paths, MPDV needs to add, for each forwarding table entry, a *capacity-source route* of the path that the entry advertises. A capacity-source route lists a path's links and the capacity of each link. Capacity-source routes are needed for two reasons: first, the source route component is used to eliminate looping paths, and second, the capacity of each link is used by the capacity removal algorithm to calculate paths.

MPDVPs

These additional elements in a router's forwarding table entries also increase the size of MPDVP's. In SPDV, a DVP contains a list of path entries where each entry consists of two elements: a path's destination address and the cost of the path. As given in Section 6.2.2, the suffix matched MPDV extension requires adding a local path identifier ϕ

RED HAT 1012
RED HAT V. CAS

to each MPDVP entry in order to ensure correct path forwarding. In addition to this local path ID, the capacity removal MPDVP entry also needs to carry the capacity-source route of the advertised path. This addition ensures that whenever a router receives a path from a MPDVP, the router has the necessary path information to perform capacity removal computation.

In summary, the additional data structure overhead of MPDV occurs in the routing forwarding tables and MPDVP messages. For the forwarding table, the extra router memory overhead is 1) capacity-source route and 2) two local path IDs for each path. To break down this cost, each path ID takes 1 byte and each element in a capacity-source route is 5 bytes (4 bytes for link specification and 1 for denoting link capacity). As for MPDV message overheads, two extra elements are added per MPDVP entry: 1) capacity-source route and 2) one local path ID for each advertised path.

The way these data structure additions are used to compute capacity removal paths is described next.

8.1.2 The MPDV Algorithm

Given the extra information in the MPDVP and forwarding table entries, each MPDV router calculates capacity removal paths. The procedure to establish suffix-matched local path IDs for path forwarding is not given here because it is described in Section 6.2.

The description of capacity removal MPDV starts with MPDVP construction. When a router constructs a MPDVP, it adds a MPDVP entry for every path it wants to advertise. A MPDVP entry consists of the path's destination address, cost, local path ID, and the path's capacity-source route. This information is contained in the router's forwarding table entry for that path. On receiving a MPDVP, a router updates the costs of each advertised path by adding the cost of the incoming link to the advertised paths; the incoming link is the link on which the router receives the MPDVP. In addition, the router appends the incoming link and the link's capacity to each MPDVP entry's capacity-source route. These two updates ensure that every path advertised in the MPDVP correctly reflects the cost and

RED HAT 1012
RED HAT V. CAS

capacity-source route with respect to the receiving router.

After updating each advertised path, a router executes the capacity removal MPDV algorithm as follows. First, the router discards any MPDVP paths that contain loops. A path loops if it contains a link in its capacity-source route that appears more than once. After discarding looping paths, the router combines, for each destination listed in the MPDVP, the paths it already has to that destination (stored in its forwarding table) with the MPDVP paths to that destination. Now that all available paths to a destination are grouped together, the capacity removal algorithm proceeds to select, for each destination, the best K paths in that group.

Selecting K paths to destination D starts by sorting all available paths to D by path cost. We refer to P_i as the path with the ith smallest cost to D and L_i as the set of links in P_i. Notice that L_i and its capacity are given in P_i's capacity-source route. Next, for each P_i, its capacity $Cap(P_i)$ is calculated as $MIN(Cap(l)), \forall l \in L_i$. If $Cap(P_i) \leq$ a preset minimum capacity threshold, then $P_i$is discarded. Otherwise P_i is admissible. If P_i is admissible, then $\forall l \in L_i, \forall j > i, \forall l' \in L_j$, if $l = l'$, then $Cap(l') = Cap(l') - Cap(P_i)$. That is, if path P_i is admissible, then subtract the capacity of P_i from all links in P_i that also appear in path P_j, $j > i$. This process starts from the least cost path to D, P_1, and ends when all paths in the group are processed or K admissible paths are found. Finally, the router installs the selected paths into its forwarding table for destination D and discards the remaining paths. This capacity removal process is repeated to find K best paths to every destination. The pseudocode of the algorithm is shown in Figure 8.1.

For each destination, the complexity of the capacity removal MPDV algorithm, as given in Figure 8.1, is linear in the number of advertised paths and the length of each path. Since there are N destinations and each router receives at most K paths per destination, the computational complexity of the MPDV algorithm is $O(NKL)$, where L is the average path length. This is a factor of KL more than SPDV, whose complexity is $O(N)$.

Notice that the paths calculated by MPDV are slightly different from the centralized capacity removal algorithm described Chapter 5. In the centralized algorithm, the ith path

RED HAT 1012
RED HAT V. CAS

```
Let A_Dst = set of advertised paths to Dst
Let F_Dst = set paths to Dst in router forwarding table
Let P_i = i^th least cost path in T_Dst

T_Dst = A_Dst ∪ F_Dst;
F_Dst = ∅;
i = 1;
while(| F_Dst |< K or T_Dst ≠ ∅)
    T_Dst = T_Dst − P_i;
    if (F_Dst == ∅ )
        Max_cost = Cost(P_i) * Cost_BOUND;
    if (Cost(P_i) > Max_cost)
        break;
    if (Cap(P_i) < Capacity_Threshold)
        continue;
    F_Dst = F_Dst ∪ P_i;
    i++;
    ∀l ∈ P_i, ∀P_j ∈ T_Dst, if (l == l' ∈ P_j)
        Cap(l') = Cap(l') - Cap(P_i);
```

Figure 8.1 : The pseudocode for the MPDV capacity removal algorithm. The code shows the calculation of K capacity removal paths to Dst.

RED HAT 1012
RED HAT V. CAS

to Dst is the ith shortest path to Dst with path capacity greater than the capacity threshold, where path i's capacity is calculated after subtracting path j's link capacities, $1 \leq j < i$. In contrast, with MPDV, R's i^{th} path to Dst is the ith shortest path advertised by R's neighbors with path capacity greater than the capacity threshold (again, after subtracting path j's link capacities, $1 \leq j < i$). Because the MPDV algorithm calculates paths based on paths advertised by neighboring routers, the MPDV algorithm always calculates suffix matched paths. In comparison, the centralized algorithm calculates its paths regardless of the paths calculated by neighboring nodes, and as a result, the centralized algorithm does not always calculate suffix matched paths. Chapter 9 shows that this slight difference in path calculation has an impact on the number of calculated paths and on the amount of offered performance.

8.1.3 MPDV Capacity Removal Example

Figure 8.2 shows an example of MPDV's capacity removal computation process. For this example, K is 3 and the minimum capacity threshold is 0. In the figure, boxes denote routers and edges denote links, and all links have unit costs. The edges are labeled by two characters: the first letter is the link's name and the second denotes the link's capacity. For example, A-2 labels link A with 2 capacity units. The dotted lines in the figure show the paths $N5$ receives from its neighbors to node $N1$.

The MPDV paths are propagated by MPDVP exchanges: initially, $N1$ sends a MPDVP to its neighbors that contains a path to itself. When $N2$ and $N4$ receive this MPDVP, they append the incoming link's label and capacity to the MPDVP entry. Specifically, $N2$ appends link A with capacity 2 to the capacity-source route to $N1$; therefore the capacity-source route of the path is [A-2]. $N4$ does the same and the capacity-source route to $N1$ becomes [C-1].* $N2$ and $N4$ then compute paths based on the new information given by $N1$'s MPDVP and propagate the newly computed paths. This example shows that $N5$ has

*Notice that because source routes are appended, the sequence of links in a capacity-source routes is the reverse of the actual paths. This does not affect the correct functioning of the algorithms.

RED HAT 1012
RED HAT V. CAS

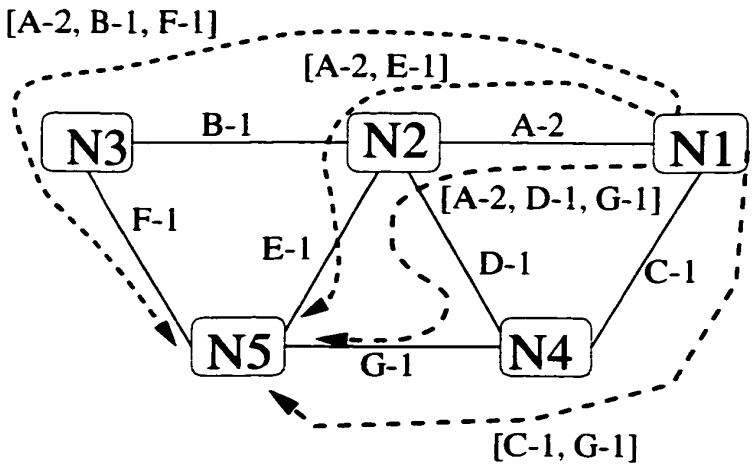


Figure 8.2 : An example of the capacity removal MPDV algorithm. The dotted lines show the paths to $N1$ that $N5$ receives from its neighbors.

received 4 paths to $N1$: (C,G), (A,E), (A,D,G), and (A,B,F). In Figure 8.2, the capacity-source route of each path is given in "[]" next to the path.

After $N5$ receives MPDVPs from its neighbors advertising these paths to $N1$, $N5$ begins the capacity removal algorithm. First $N5$ sorts the paths by cost: (G,C), (E,A), (G,D,A), (F,B,A). Next, $N5$ inspects each path in ascending order of path cost. The first path, (G,C), is admissible because $Cap(C, G) = 1 > 0$. Therefore the link capacities of C and G are subtracted by $Cap(C, G) = 1$. The path (E,A) is also admissible, and the capacities of links E and A are also subtracted by 1. Next, $N5$ considers the path (G,D,A). Here, $Cap(G, D, A) = 0$ because the capacity of link G is 0 (link G's capacity was subtracted by 1 from path (G,C)). Since $Cap(G, D, A) \leq 0$, this path is discarded. Finally, path (F,B,A) is admissible with capacity 1. With $K = 3$, $N5$ keeps paths (G,C), (E,A), and (F,B,A) in its forwarding table. Now whenever $N5$ advertises paths to $N1$, it puts these three paths into its MPDVP.

RED HAT 1012
RED HAT V. CAS

8.1.4 Capacity Removal MPDV Costs

Given the capacity removal MPDV algorithm, this subsection discusses the algorithm's costs. The MPDV costs are categorized by 1) per packet forwarding overhead, 2) router CPU usage, 3) routing message, and 4) router memory overhead.

The first and the most performance critical overhead is the per packet path forwarding overhead. This overhead consists of the additional per packet path ID cost (in the number of bits) and additional router processing needed to interpret path IDs and to forward packets. Because of the suffix matched forwarding method, MPDV's per packet overhead of path specification is constant and fixed-length. Moreover, the suffix matched path IDs are small integers, thereby allowing efficient router forwarding table lookup which decreases the per packet forwarding time [149, 160]. With low per packet forwarding overhead, MPDV decreases the time to transmit and forward data packets.

The second and third MPDV cost categories are router CPU usage and routing messages. With respect to the message complexity, the previous section showed that each MPDVP contains at most $O(NK)$ entries where each entry is length $O(L)$. Thus the message complexity of each MPDVP is $O(NKL)$, a factor of KL more than that of SPDV.

The message complexity of the entire MPDV algorithm is the product of the complexity of individual MPDVP's and the number of MPDVP's sent. Although it has been shown that the worse case number of messages for a Distance Vector style algorithm is $O(2^N)$ [25], the average case complexity is $\Theta(NM^3(ln(M))^2)$ [23,157]; here M is the average number of neighboring routers. Therefore the average message complexity of the capacity removal algorithm is $\Theta(NKL * NM^3(ln(M))^2) = \Theta(N^2M^3KL(ln(M))^2)$. Again, this is a factor of KL more than that of SPDV.

Moreover, because DV style algorithms exchange paths via messages, the computational complexity is the same as the message complexity. That is, the average MPDV computational complexity is also $\Theta(N^2M^3KL(ln(M))^2)$.

With respect to MPDV's router memory requirement, MPDV routers need memory proportional to the number of paths calculated and their length, $O(KNL)$. This is compared to

RED HAT 1012
RED HAT V. CAS

SPDV where memory and message overheads are proportional only to the number of paths calculated $O(N)$. However, it is worth noting that although MPDV requires KL additional memory, this does not mean that the forwarding tables are KL times larger (for simplicity in our previous discussion, we used the term "forwarding table overhead" synonymously with "router memory overhead"). The reason is that the capacity-source routes are not needed for packet forwarding; therefore these source routes can be stored elsewhere, say in slower and cheaper router memory. Thus, although the MPDV message overhead is KL more than SPDV, the performance critical forward table overhead is actually only a factor of K.

In summary, capacity removal MPDV overheads in router memory, router CPU, and routing messages are a factor KL more than SPDV. In addition, the performance critical cost category, per packet forwarding overhead, is low because the suffix matched forwarding method uses small integer, fixed-length path IDs, allowing efficient packet transmission and forwarding.

8.2 The MPLS Capacity Removal Algorithm

This section describes the implementation of an LS based multipath routing algorithm that calculates capacity removal paths. The LS routing algorithm is the basis of many widely used routing algorithms. Unlike DV, LS routers share topology information and each router computes paths with the knowledge of the entire network graph. As the implementation shows, this style of path computation has a large impact on the amount of resources LS needs to compute multiple paths.

8.2.1 The MPLS Algorithm

Before describing the MPLS algorithm, we briefly review of the traditional single path LS (SPLS) algorithm. In SPLS, each router periodically broadcasts its local topology in a Link State Packet (LSP). This information consists of the router ID and the cost of each of its out-going links. Routers gather these broadcasts, locally construct the entire network

RED HAT 1012
RED HAT V. CAS

graph, and perform a shortest path computation to all other nodes. For each router, the result of the path computation is stored in its forwarding table.

To convert SPLS to MPLS that compute capacity removal paths is very straightforward. First, the LSP is augmented with a capacity component: an LSP now consists of the sending router's ID plus the cost and capacity of each of the router's out-going links. MPLS routers periodically broadcast their LSPs as in the SPLS algorithm. After all routers broadcast their LSPs, each router constructs the entire network graph knowing the cost and capacity of each network link. After collecting this information, routers individually execute the centralized capacity removal path calculation algorithm.

The centralized capacity removal algorithm is described in Section 5.2.2. To summarize, the algorithm finds successive shortest paths to destinations using only links that are above a capacity threshold. When the current shortest path is found to a destination, the algorithm tests if it exceeds the cost bound. If so, path computation for that destination ends. Otherwise, the path is stored in that destination's forwarding table, and the path's capacity is subtracted from each link in that path. This process continues until K paths are found or until no other admissible path exists. In this chapter's implementation, the cost bound is set at twice the cost of the shortest path, and the capacity threshold is zero.

The time complexity for the LS capacity removal algorithm to calculate K paths to a destination is $O(K * E * lg(E))$, where E is the number of edges in the network.

8.2.2 Basic MPLS Forwarding

The previous subsection described the MPLS capacity removal computation, but it did not describe the packet forwarding on the computed paths. Like MPDV, this implementation of the capacity removal MPLS algorithm uses the suffix matched forwarding method. However, unlike MPDV where establishing local path IDs is straightforward, establishing path IDs in MPLS may require additional computation and messages, depending on the type of paths computed. The basic MPLS suffix matched forwarding method is given here, and extensions to cover non-suffix matched paths are given in the next subsection.

RED HAT 1012
RED HAT V. CAS

The essence of the suffix matched forwarding method is the establishment of local path IDs between nodes. These local path IDs guarantee forwarding of packets on their intended paths. In MPLS, establishing local path IDs is a three-step process. In the first step, every router labels the ith path to destination D with the path ID i, $i \leq K$. Second, for each path $(x_1, \ldots, x_n)$ that x_1 computes, x_1 finds the path ID of the path $(x_2, \ldots, x_n)$ as calculated by x_1's neighbor x_2. That is, router x_1 finds x_2's j^{th} path to x_n such that the jth path is $(x_2, \ldots, x_n)$. This calculation is very simple because x_1 has the entire network topology in its memory, and the path ID assignments are deterministic (i.e. x_1 and x_2 will both calculate the path $(x_2, \ldots, x_n)$ as x_2's j^{th} path to x_n).

In the third step, x_1 matches the local path ID of the path $(x_1, \ldots, x_n)$ with x_2's path ID of the path $(x_2, \ldots, x_n)$ and stores the two path IDs in the forwarding table entry for path $(x_1, \ldots, x_n)$. Now that the two path IDs are established on this path, suffix matched path forwarding can be used. This process of calculating and matching path IDs is executed by every router for every calculated path. For more details on the MPLS suffix matched forwarding method, refer to Section 6.3.

8.2.3 Non-Suffix Matched Paths

The three step process described above correctly forwards packets if the path set is suffix matched. Unfortunately, the centralized capacity removal algorithm does not always compute suffix matched paths. This means that in step 2, when a router attempts to find the a match for a non-suffix matched path, it will discover that the path's next-hop neighbor does not calculate the suffix of that path. Fortunately, there are techniques to resolve these non-suffix matched paths; a list of techniques is provided in Section 6.3. The one used in here is the explicit routing method. We choose this method because of its efficiency and because we can explicitly measure its message and storage costs.

The explicit routing method operates as follows: during step 2 of path ID matching, if a router fails to match the suffix of a path, the router stores the source route of this non-suffix matched path. When a router receives a packet with a path ID specifying the non-suffix

RED HAT 1012
RED HAT V. CAS

matched path (indicated by the source route), the router source routes the packet to its destination. Furthermore, explicit routing uses the label swapping optimization to establish local path IDs after source routing the first packet [85]. Therefore, subsequent packets traversing the same path will be forwarded using the conventional fixed-length path IDs. This optimization significantly reduces the source routing cost because it is done only *once* per non-suffix matched path. Although, these labels need to be re-established when paths are recalculated (say due to link failures), the amortized cost of source routing packets on non-suffix matched paths is small because path recomputation occurs much less frequently compared to packet forwarding. An example of the explicit routing method is given in Figure 8.3.

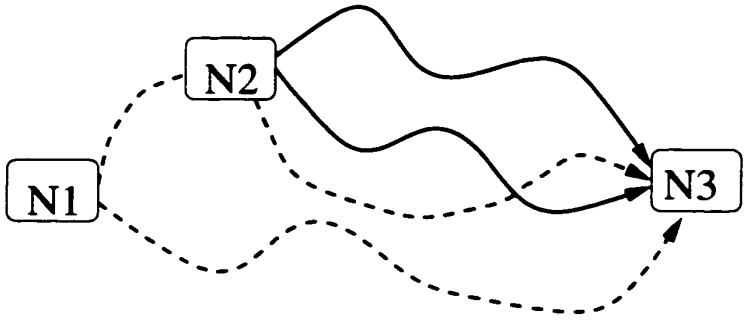


Figure 8.3 : An example of the capacity removal MPLS non-suffix matched forwarding. Here N1 computes 2 paths to N3 denoted by the dotted lines, and N2 computes the two solid paths to N3. N1's paths through N2 are not suffix matched because N2 did not compute a path that is the suffix of N1's path.

In this figure, both N1 and N2 compute two paths to N3, denoted by the dotted and solid lines respectively. Notice N1's path that passes through N2 is not suffix matched because N2 did not compute the suffix of N1's path. According to the explicit routing method, upon receiving the first packet destined on this non-suffix matched path, N1 will source route this packet on the dotted path through N2 to N3. In addition, this first packet will establish local path IDs along routers on this path so that subsequent packets destined for this path will not be source routed, but instead, use the path ID established by the source-routed

RED HAT 1012
RED HAT V. CAS

packet. Thus, subsequent packets on non-suffix matched paths will be forwarded in the same manner as packets on suffix matched paths.

8.2.4 Capacity Removal MPLS Costs

For the most part, the implementation of the capacity removal MPLS algorithm is straightforward. This section summarizes the various costs incurred by the algorithm: 1) per packet forwarding overhead, 2) router CPU usage, 3) routing message, and 4) router memory overhead.

Like MPDV, the performance critical per packet forwarding overhead of MPLS is very efficient: fixed-size per packet path ID which can be efficiently indexed in router forwarding tables. However, unlike MPDV where all paths are suffix matched, some paths in capacity removal MPLS are not. For these paths, source routing is required. Since the label swapping optimization is used, source routing is incurred only once per path, and non-suffix paths need to be re-source routed only after path recomputation, which occurs infrequently compared to data packet forwarding. Thus the cost of source routing does not significantly contribute to packet forwarding complexity.

In the second cost category, router CPU overhead, the time complexity of the centralized capacity removal algorithm to compute all pairs K paths is $O(KNE * lg(E))$. Since each router also needs to calculate paths for its neighbors, the time complexity is $O(KNEM * lg(E))$, where M is the average number of neighboring routers. This is compared to shortest path LS whose complexity is $O(E * lg(E))$. Although the CPU complexity is a factor KNM more, we believe this computation overhead can be offset by leveraging the technology curve (e.g. faster processors and larger memories) [11] and developing more efficient capacity removal algorithms that use dynamic programming.

The routing message cost is divided into two categories: the extra cost in the LSPs and the costs incurred by source routing packets on non-suffix matched paths. For the first category, the cost overhead is nominal – one capacity specification per link in an LSP. In this implementation, this adds only one additional byte per link in a LSP; thus the MPLS's

RED HAT 1012
RED HAT V. CAS

LSP complexity is the same as single path LS, $O(NE)$. The second message cost category, source routing cost, depends on the number of non-suffix matched paths and the length of each path. However, the use of the label swapping optimization significantly decreases the source routing cost. Given that the source routing message cost depends on the number of non-suffix matched paths and their length, this cost is experimentally quantified in Section 9.4.

With respect to router memory, there are two types of router storage: general purpose memory and forwarding table memory. As with MPDV storage, the memory requirement for the general purpose memory is not as crucial as the forwarding memory, which is performance critical.

A MPLS router's general storage complexity is typically dominated by storing the entire topology in memory: each router needs to store every other router's LSPs, incurring $O(NM)$ memory costs. With non-suffix matched paths, an MPLS router also needs to store the non-suffix matched source routes, requiring an additional $O(NpL)$ storage, where p is the average number of non-suffix matched paths to a destination passing through a router, and L the average path length.

With respect to forwarding table storage, the MPLS cost is proportional to the number of paths calculated. However, due to non-suffix matched paths, a router may store paths that pass through the router but which the router itself does not compute. For example, router N2 in Figure 8.3 will need to store three paths, even though it only calculated two. This additional path entry comes from establishing a local path ID with N1 for N1's non-suffix matched path. Thus, the router forwarding table storage complexity for capacity removal MPLS is $O(KN + NpL)$.

Because router storage costs depend on the number of non-suffix matched paths and their length, this cost is experimentally quantified in the next chapter.

In general, the routing costs for the capacity removal MPLS routing algorithm are low, and because MPLS has the entire network topology in memory, calculating multiple paths is straightforward. However, there are costs in MPLS that cannot be analytically deter-

RED HAT 1012
RED HAT V. CAS

mined, specifically, the message cost of source routing packets and router storage costs. For these costs that depend on non-suffix matched paths, the next chapter experimentally measures their actual costs.

RED HAT 1012
RED HAT V. CAS

Chapter 9

Multipath Cost-Benefit Analysis

In this chapter, we experimentally evaluate the performance improvements obtained by using capacity removal MPDV and MPLS algorithms, and then measure each routing algorithm's message, computation, and storage costs. The goal of these measurements is to determine whether the cost incurred by a multipath routing algorithm justifies the end-to-end performance gains obtained using the provided paths.

The performance is measured in terms of throughput, latency, and message drop probability (message delivery reliability). Throughput is measured using MPTCP. The higher the MPTCP obtained throughput, the better throughput performance of the multipath network. Latency and drop probability are measured using a multipath ping program. This program uses multiple paths to measure round-trip delay and drop probabilities. To make the measurements more realistic, the ping experiments are conducted using the background traffic generated by the throughput experiments.

In addition to measuring the performance offered by a multipath network, this chapter also measures the performance (or efficiency) of the routing algorithms that implement multipath networks. The two algorithms, capacity removal MPLS and MPDV, are evaluated in this chapter in terms their packet forwarding cost, router memory requirement, router CPU usage, and routing message cost. Although many of the routing costs are analytically evaluated, some routing costs depend on the network topology and the paths computed between nodes. This chapter experimentally measures these network dependent costs, as well as the constants of the analytically formulated costs.

The results of the experiments show that even in sparse topologies (a 100 node, Internet-like cluster topologies with link to node ratio less than 2:1), the capacity removal MPDV

RED HAT 1012
RED HAT V. CAS

and MPLS are able to provide paths that allow both MPTCP and multipath ping to achieve higher performance, compared to their single path counterparts. Moreover, the costs incurred by MPLS and MPDV are low enough such that we believe they can be feasibly implemented in large-scale networks.

This chapter is organized as follows. The next subsection describes the experimental environment. Section 9.2 presents MPTCP throughput, and Section 9.3 presents the latency and drop probability performance. Finally, Section 9.4 presents the various routing costs of the two routing algorithms. A summary of the experiments appears in Section 9.5.

9.1 Simulation Environment

The simulation environment consists of the network simulator and the network topology. The network simulator used is based on the "xsim" package from the University of Arizona [28], an execution-driven, packet-level network simulator. The simulator takes as input a description of the network topology, including link characteristics such as bandwidth and propagation delay, and a set of software modules that implement the various protocols running on the routers and end-hosts in the network. Simulation time advances according to the calculated transmission and propagation delay of packets in the network. In the simulator, software processing in the routers and hosts is assumed to have zero time cost.

Software processing occurs in three places in our simulation: in end-host processing, router path computation, and router packet forwarding. In each of these places, we believe this zero time cost assumption does not significantly affect our results. First, end-host processing is usually several orders of magnitude faster than end-to-end network latency; therefore accounting for their processing costs has little impact on end-to-end measured performance. Second, since router path computation is *not* performance critical, occurs infrequently compared to packet forwarding, and the computational complexity of our MPDV and MPLS implementation are based on their single path counterparts, accounting their computational time should not significantly alter our performance measurement. And third, with respect to the performance critical cost of router packet forwarding, our

RED HAT 1012
RED HAT V. CAS

multipath forwarding method adds only one additional hash lookup in the forwarding process; therefore, the additional processing cost of forwarding is also very small and should not have a large impact on our simulation results.

To make the performance and cost measurements more realistic, the experiments in this chapter use an Internet-like clustered topology. Cluster topology is created by first constructing a group (or cluster) of small *flat topologies*. Flat topologies are parameterized by the number of nodes (N) and links (E). The construction process randomly picks two nodes and connects them until all E links have been connected. The only restriction on node connection is that no more than one link can exist between a node pair. After connecting E links, the network is inspected for connectedness. If the network is connected, a flat topology is generated and returned. Otherwise, the network is discarded, and the flat topology construction process is repeated.

A clustered network topology is build by inter-connecting a set of flat topologies. The flat topologies are connected by randomly connecting nodes from different flat topologies (clusters). Links connecting nodes within a cluster (intra-cluster links) have 1000 KB/s bandwidth and unit cost, and links connecting clusters (inter-cluster links) have 2000 KB/s bandwidth and two units of cost. All links have a 10us transmission delay. These parameters where chosen to reflect the higher connectivity within clusters and the larger bandwidth of inter-cluster links.

9.2 Throughput Performance

This section evaluates the throughput offered by the capacity removal MPDV and MPLS routing algorithms. Throughput is measured by end-hosts transmitting packets using MPTCP on networks running each routing algorithm. Recall Chapter 7 showed that the capacity removal algorithm effectively computes paths with additional network resources, and that the MPTCP protocol is able to translate the additional resources into increased throughput. This section extends the results of Chapter 7 and investigates whether these throughput improvements can actually be obtained using the capacity removal MPLS and MPDV routing

RED HAT 1012
RED HAT V. CAS

algorithms.

9.2.1 Basic Throughput Performance

To determine whether MPTCP is able to achieve the same level of performance using MPLS and MPDV, the first experiment conducted in this section uses the same experimental parameters as the MPTCP experiment in Section 7.5.

In summary, the experimental network consists of 100 nodes organized in 10 node clusters with 17 intra-cluster links and 25 inter-cluster links (a total of 195 links).* Traffic is divided into two categories: foreground traffic and background traffic. A foreground traffic node transmits 5,000 packets to a node in another cluster, and a background traffic node sends a burst of packets ranging from 100 - 2,000 packets to a destination chosen at random. The size of each packet is 1500 bytes. The inter-burst times of background nodes are exponentially distributed (a Pareto distribution with average of 20 seconds). The three foreground nodes are picked randomly and the rest are background nodes. The performance results of MPTCP on both MPDV and MPLS are given in Figures 9.1 and 9.2.

The graphs in Figures 9.1 and 9.2 show the MPTCP foreground and background performance on MPLS and MPDV respectively. In both figures, the left graph shows the foreground performance curves, and the right graph shows background performance. The x-axis denotes background traffic burst sizes, ranging from 100 to 2,000 packets, and y-axis denotes the effective throughput in KB/s for the respective foreground or background traffic. Curves in each figure are labeled to represent the transport protocol used by foreground and background nodes: the first label denotes the foreground transmission style and the second the background transmission style. SP stands for using SPTCP, and MP2 and

*This network topology is particularly desirable because it is not too connected and not too sparse. The network is not too connected so that the traffic generated will produce network contention in order to show interesting properties of multipath routing, and the network is not too sparse so that meaningful multiple paths can be calculated between nodes. In short, the 1:2 node to link ratio highlights performance differences in single path and multipath routing.

RED HAT 1012
RED HAT V. CAS

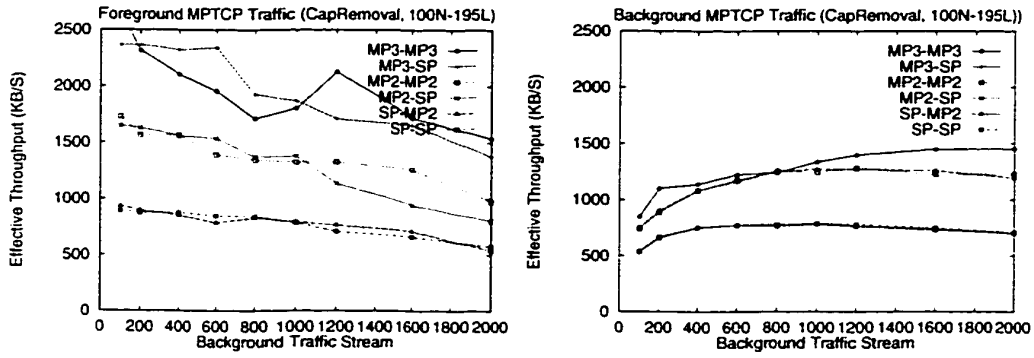


Figure 9.1 : The foreground and background MPTCP performance using the MPLS capacity removal algorithm (same as Figure 7.5).

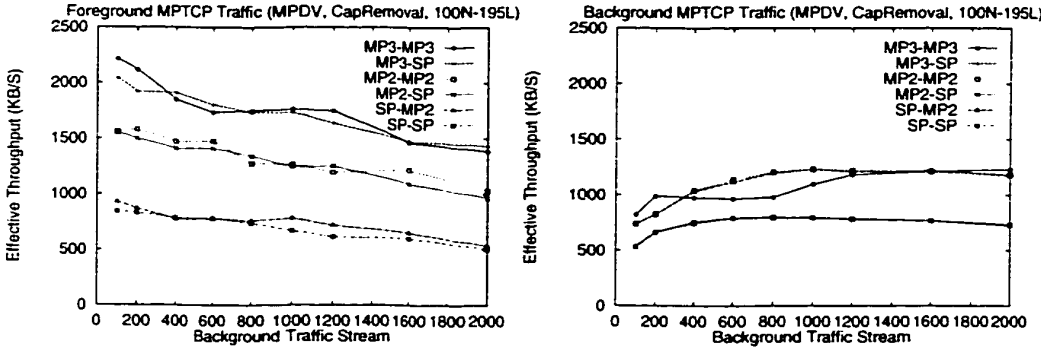


Figure 9.2 : The foreground and background MPTCP performance using MPDV capacity removal algorithm.

MP3 stand for using MPTCP with 2 and 3 paths respectively. For example, the curve labeled MP3-SP represents the network performance when the foreground nodes used 3-path MPTCP and the background nodes used SPTCP.

The graphs in Figures 9.1 and 9.2 show that foreground and background MPTCP performances exhibit the same characteristics described in Chapter 7. Mainly, MPTCP is effective in increasing network throughput even at high levels of network utilization, and the performance gains of multipath MPTCP do not come at the expense of SPTCP connections. For detailed MPTCP analysis, refer to Section 7.5. In addition to MPTCP behavior, two other important points are illustrated in the two figures.

RED HAT 1012
RED HAT V. CAS

First and foremost, both figures show that MPTCP obtains a high level of performance with both MPLS and MPDV. Moreover, the graphs show that MPTCP foreground performance is roughly proportional to the number of paths provided by each algorithm. This demonstrates that MPLS and MPDV algorithms are able to provide paths with additional resources, thereby allowing MPTCP to improve its throughput. The two figures confirm that in a realistic network setting, the combination of MPTCP and capacity removal MPDV/MPLS algorithms achieves higher network throughput than is achievable in single path routing.

The second important result in Figures 9.1 and 9.2 is that MPTCP performance is higher in MPLS than in MPDV. Performance using MPLS is the same as offered by the centralized capacity removal algorithm shown in Figure 7.5 because MPLS calculates paths using the same centralized capacity removal algorithm. In contrast, MPTCP's performance on MPDV is worse than on MPLS. The performance difference is due to the differences in MPLS and MPDV path calculation. Specifically, MPDV always calculates suffix matched paths, and MPLS does not.

In the following three subsections, we investigate the MPLS and MPDV performance disparity caused by their path calculation differences.

9.2.2 MPLS and MPDV Throughput Differences

To investigate the impact of MPLS and MPDV path computation differences, the same simulation was rerun in a sparser topology to reduce the number of available physical paths, thereby emphasizing the differences in path computation styles. This sparse network has 15 intra-cluster and 20 inter-cluster links (170 links), as compared to the original network which has 17 intra-cluster and 25 inter-cluster links (195 links). Figure 9.3 shows the MPTCP performance results in this simulation environment.

To highlight performance differences, the graph in Figure 9.3 shows only MPTCP foreground traffic performance in the sparse network. For each curve, the background nodes use the same transmission style as the foreground nodes. The graph shows the MPTCP

RED HAT 1012
RED HAT V. CAS

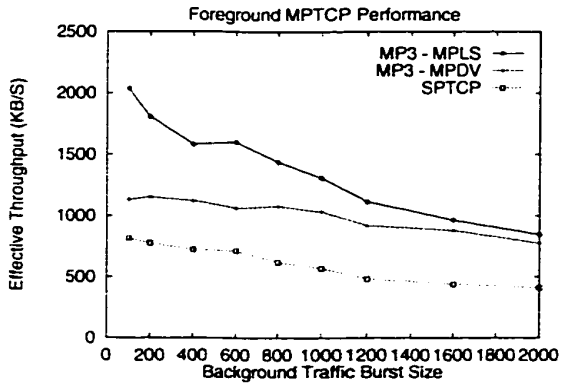


Figure 9.3 : The foreground performance of MPTCP and SPTCP using the capacity removal MPLS and MPDV algorithms on a sparse 100 node, 170 link network topology.

performance using 3-path MPLS, 3-path MPDV, and SPTCP. Three-path MPTCP is featured because using more paths accentuates the performance differences between MPLS and MPDV. The background performances are not shown because they exhibit the same performance characteristics as in Figures 9.2 and 9.1.

As with the previous topology, Figure 9.3 shows that MPTCP is able to obtain multipath performance benefits using MPDV and MPLS. This shows the generality and versatility of MPTCP and MPLS/MPDV in increasing network throughput even in a topology where the ratio of links to nodes is less than 2:1. In addition, the figure shows that MPTCP foreground performance decreases as the amount of background traffic increases. This is again consistent with our observation that as total network traffic increases, average individual throughput decreases.

The notable feature illustrated by Figure 9.3 is the MPTCP performance difference between MPDV and MPLS. MPTCP performs better under MPLS than under MPDV, especially in the region where network utilization is low. Analysis of the number of paths calculated by each routing algorithm reveals the source of this performance disparity. Figure 9.4 shows the number of paths calculated by MPLS and MPDV in this network.

In Figure 9.4, the x-axis denotes rank, the maximum number of paths the routing algorithm is allowed to calculate between a node pair, and the y-axis shows the total number

RED HAT 1012
RED HAT V. CAS

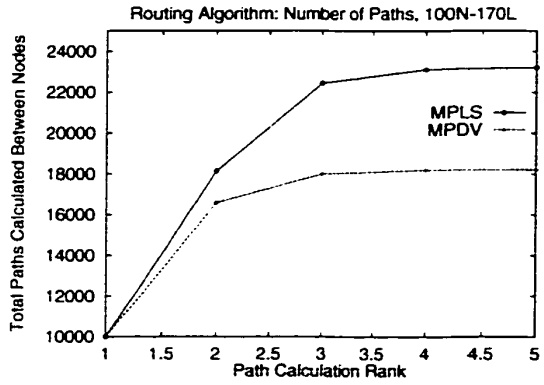


Figure 9.4 : The number of path calculated by each the MPDV and MPLS algorithms in the 170 link sparse network topology.

of paths actually calculated. At rank = 1 ($K = 1$), both algorithms calculate the single shortest paths. Since there are 100 nodes, the number of calculated path is $100^2 = 10,000$ paths.

As the figure shows, the difference between the paths calculated by MPLS and MPDV increases with rank. Moreover, this increase is sub-linear because the number of possible paths is limited by the physical connectivity of the network. The figure shows that at rank 3, MPDV calculates 18000 and MPLS calculates 22000 paths, a difference of 22%. These extra MPLS paths allow MPTCP to achieve higher throughput.

9.2.3 MPLS and MPDV Path Calculation Process

MPLS calculates more paths than MPDV because MPLS can calculate paths that are not suffix matched. In other words, MPDV is restricted in its path calculation because it always calculates suffix matched paths. Figure 9.5 shows how this restriction influences the paths that an algorithm can calculate.

In Figure 9.5, the boxes denote routers and curves denote paths. Here, capacity removal paths to N1 are being calculated. Assume that N2's shortest path to N1 is the dotted path labeled A, and that N3's shortest path to N1 is the solid path C. Assume further that rank = 2, and all links have equal capacity (i.e. calculating link disjoint paths). In the MPLS

RED HAT 1012
RED HAT V. CAS

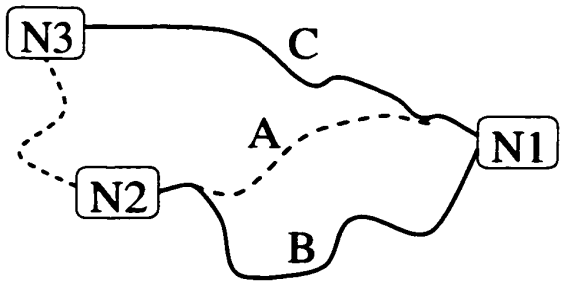


Figure 9.5 : An example of the capacity removal path calculation to $N1$.

algorithm, N3 will calculate two paths: the first path is C, and the second one passes through N2 via the dotted curve and connects to path B. Using MPDV, however, N3 will calculate only one path. This is because N2 will select and propagate only path A and discard path B, since path B is inadmissible because it has common links with path A. So when N3 receives N2's advertised path A, N3 discards A because A is inadmissible due to the links shared with path C.

The calculation of non-suffix matched paths is precisely the reason MPLS calculates more paths than MPDV. Recall, a path set P is suffix matched if and only if $\forall \alpha \in P, \alpha = (x_0, \ldots, x_n)$, then $(x_i, \ldots, x_n) \in P, \forall i, 0 \leq i < n$. Figure 9.6 shows the percentage of non-suffix matched paths that MPLS calculates in this sparse topology. The percentage of non-suffix matched path is the number of non-suffix matched paths divided by the total number of paths calculated.

The Figure 9.6 shows that at rank = 3, approximately 35% of the paths calculated by MPLS are non-suffix matched. Because of these paths, MPLS is able to calculate approximately 22% more paths than MPDV (Figure 9.4). These extra paths, in turn, contribute to higher MPTCP performance.

9.2.4 MPLS and MPDV Throughput Summary

This section analyzed the throughput of MPLS and MPDV capacity removal routing algorithms. The experiments show that even with relatively sparse topologies (100 nodes

RED HAT 1012
RED HAT V. CAS

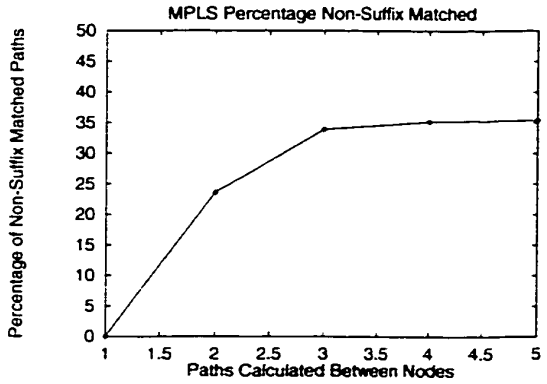


Figure 9.6 : The percentage of non-suffix matched paths calculated by MPLS in a 100 node, 170 link cluster topology.

with 195 links and 170 links), MPLS and MPDV were able to calculate paths that allow end-hosts to increase their throughput. Our experiments demonstrate that the capacity removal MPLS and MPDV algorithms, in conjunction with MPTCP, substantially increase end-to-end throughput compared to TCP using single path routing.

Further examination of MPLS and MPDV shows that under certain circumstances, MPDV does not compute as many paths as MPLS. The reason is that capacity removal MPLS can calculate non-suffix matched paths, but capacity removal MPDV cannot. This difference in path computation results in lower MPDV performance in scenarios where calculating only suffix matched paths produces a significantly lower number of paths. The performance impact of this disparity is reflected in MPTCP's achievable throughput.

9.3 Latency and Message Drop Performance

The previous section showed that multipath routing can be of immediate benefit to today's applications by offering increased network throughput. The purpose of this section is to show that multipath routing can also be used to reduce end-to-end network latency and message drop probability (or increase likelihood of message delivery), two performance metrics also of immediate benefit for today's applications.

To measure these two performance metrics, this section uses a multipath ping program

RED HAT 1012
RED HAT V. CAS

that measures observed round-trip latency and message drops. Round trip latency is an important performance metric for interactive applications such as telnet and web sessions, and message drop probability is used for applications that want to increase the reliability of their message transmission.

To provide a realistic traffic scenario, the multipath ping program is run in the presence of the background traffic generated in the throughput experiments. Simulation results show that under all traffic conditions, the multipath ping program achieves lower round-trip delay and a lower number of message drops compared to a single path ping program. The conclusion of our experiments is that in addition to providing increased throughput, multipath routing *simultaneously* allows decreased round-trip delays and message drop probabilities.

9.3.1 The Multipath Ping Program

The operation of a ping program is very simple: given a destination address, the program sends a message to the destination, and upon receiving this message, the destination's ping program sends the message back to the sender. Round-trip latency is then calculated by subtracting the time the sender sends the message by the time the sender receives the reply message.

The single path ping program does exactly as described in the ping program specification: the sender sends a ping message to its destination; the destination then echos a ping upon receiving the ping message. However, because the multipath ping program can send messages of more than one path, multipath ping can selectively choose the lowest delay path to the destination, and similarly, the destination can choose the lowest delay path back to the sender. The multipath ping program developed in this section uses the minimal delay path to and from destinations.

The multipath ping program operates as follows. Given a destination address, the sender duplicates the ping message on all available paths to that destination. Upon receiving the first ping message, the destination immediately sends a reply message on all available paths back to the original sender. The destination ignores subsequent ping messages that arrive on

RED HAT 1012
RED HAT V. CAS

different but slower paths. When the original sender receives a destination's reply (the first one), it calculates the round-trip time. The sender also ignores the subsequent destination reply messages that arrive on slower paths. If the sender does not receive any ping message replies, the message is counted as a dropped message.

Notice that in addition to minimizing delay, the multipath ping approach also reduces the probability that a ping is dropped by duplicating messages on multiple paths. Although this duplication approach may not be appropriate for many interactive applications (because it sends more messages), there are applications that need to send urgent messages and want these messages to have minimum transmission time and message drop probability.

For those applications that wish to lower communication delays without message duplication, they (or a low latency protocol) can periodically "ping" all available paths and then solely use the path that provides the lowest measured delay. For example, a possible low latency protocol could duplicate only retransmitted messages on multiple paths. This approach not only urgently retransmits a dropped message, but also uses this retransmitted message to find the current least delay path. The newly selected least delay path is then solely used for subsequent message delivery, until the next message drop.

For the purpose of this experiment, the multipath ping protocol described above measures the appropriate performance metrics: the reductions in round-trip latency and message drops that a multipath network can provide. The simulation results are presented next.

9.3.2 Round-trip Latency

Both the multipath and single path ping programs are implemented in our network simulator. The experimental configuration used to measure their performance is the same as the one used in the first throughput experiment: the 100 router, 195 link cluster network topology, MPTCP and SPTCP generated traffic, and the MPLS capacity removal routing algorithm (the performance results capacity removal MPDV are similar and not shown here).

In the latency experiment, each ping host sends a ping message to a randomly chosen

RED HAT 1012
RED HAT V. CAS

destination. The time between a host's transmission of two consecutive pings is exponentially distributed (a Pareto distribution with average of 5 seconds). The latency and the percentage of dropped pings of single path ping (SP_ping) and multipath ping (MP_ping) are given in Figures 9.7 and 9.8.

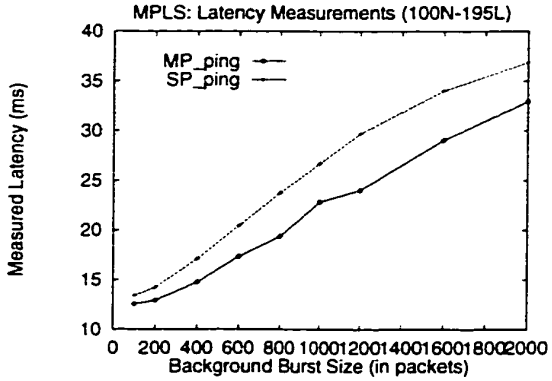


Figure 9.7 : The measured round-trip latency observed by the MP_ping and SP_ping programs. The experiment is conducted using a 100 node, 195 link cluster network.

The graph in Figure 9.7 shows two latency measurements: the latency of 3-path MP_ping on a multipath network where the background traffic uses 3-path MPTCP, and the latency of SP_ping where the background traffic uses SPTCP. In this figure, the x-axis denotes the background traffic burst sizes, and the y-axis denotes the average latency observed by the each ping program. Notice that the 3-path MPTCP generates more background traffic than SPTCP. That is, for the same background burst size (x-axis), the multipath network provides higher overall throughput than the single path network. In addition to offering higher throughput, this experiment shows that multipath routing also provides lower round-trip latency and message drop probability.

The latency graph in Figure 9.7 shows that as the background traffic increases (indicated by increasing the burst size), the round-trip latency increases correspondingly. This is expected because higher network traffic increases router queuing, which in turn increases network delay. In addition, the graph shows that as network traffic increases, the disparity in round-trip latency between MP_ping and SP_ping also increases. The reason is that

RED HAT 1012
RED HAT V. CAS

at low network traffic, all paths have relatively low queuing delays, so the delay difference between the minimal delay path and the static metric shortest path is relatively small. However, at high network traffic, the delay differences among multiple paths increase; hence the increase disparity in SP_ping and MP_ping round-trip latencies.

9.3.3 Message Drop Probability

The second performance metric measured message drop probability. Message drop probability is defined as the likelihood that a transmitted message is *not* received at its destination: if a message is duplicated, then the message drop occurs when all duplicates are dropped by the network and never reaches the intended destination.

The multipath ping program is used to measure message drops. Because ping messages and their replies are sent on multiple paths, a message drop occurs only if *every* path to or from the destination drops the ping or reply ping messages. That is, assuming equal likelihood of packet drops, if the drop probability of one path is p, then drop probability of SP_ping is $2 * p$ (a message has to travel to and from the destination), and the drop probability of MP_ping is $2 * p^K$, where K is the number of paths between nodes. The drop measurements for SP_ping and MP_ping are given in Figure 9.8.

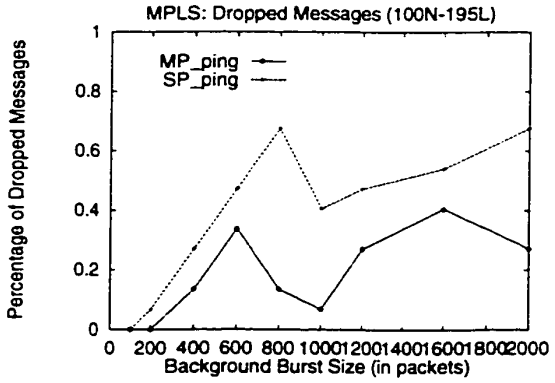


Figure 9.8 : The measured ping drop percentages of MP_ping and SP_ping. The results shown are collected in the same experiments that produced the latency measurements in Figure 9.7.

RED HAT 1012
RED HAT V. CAS

The graph in Figure 9.8 shows the percentage of dropped pings observed by SP_ping and MP_ping. The x-axis shows background traffic burst sizes, and y-axis shows the percentage of dropped ping messages. The results shown in this figure are gather in the same simulations that produced the latency measurements in Figure 9.7. The graph confirms our analysis that the MP_ping program incurs less message drops compared to SP_ping. At high network utilization levels (e.g. background burst sizes $\geq$ 600 packets), MP_ping incurs approximately half the drops compared to SP_ping.

However, given that MP_ping uses three paths, one expects the MP_ping to drop messages more seldom than 50% of SP_ping drops. One factor causing higher MP_ping drop rate is that MP_ping's background nodes (which use 3-path MPTCP) generate, according to Figure 9.1, approximately two times more traffic than SP_ping's background nodes (which use single path TCP). Because of the higher traffic rates, the probability, p, that a path will drop a packet in a multipath environment is higher than in a single path environment.

The spikes and dips of message drop percentages in Figure 9.8 are a product of random traffic generation. Because the initiation times of ping messages and background traffic use a random distribution, the spikes are the result of a combination of an unusually high traffic generation in the presence of ping measurements. Similarly, the dips are the result of the opposite scenario. The two curves have similar shape because their traffic use the same random distribution seed. However, since MPTCP nodes achieve higher throughput (thereby reducing transmission times), the background traffic sending times slightly differ between TCP and MPTCP sending nodes; thus the two curves do not have precisely the same shape.

9.3.4 Latency and Drop Probability Summary

This section shows that in addition to increasing throughput, a multipath network can also increase application performance through decreased round-trip latency and message drop probability. To quantify these performance improvements, a multipath ping program is developed that uses multiple paths to minimize round-trip latency and message drops.

RED HAT 1012
RED HAT V. CAS

Using a 100 node Internet-like cluster network and realistic MPTCP/TCP background traffic, we measure the performance of the single path (SP_ping) and multipath ping (MP_ping) programs. Simulation results show that not only do multipath networks increase throughput compared to single path networks (Section 9.2), multipath networks *simultaneously* allow the MP_ping program to achieve lower round-trip delay and lower drop probability compared to the SP_ping program on a single path network.

The conclusion of our performance experiments is that the additional resources offered by a multipath network can be effectively used to increase network performance, measured in throughput, latency, and message drop probability.

9.4 Routing Costs

The previous sections show the improvements in throughput, latency, and message drops offered by the capacity removal MPDV and MPLS routing algorithms. This section examines the overhead incurred by the two routing algorithms. The runtime cost of MPDV and MPLS, as stated in Section 4.2.3, consists of three components.

1. Packet forwarding: overhead of per packet path specification and additional router processing needed to deliver packets on multiple paths.

2. Router Storage: additional memory needed to store and support multiple paths.

3. Path computation: routing and algorithmic overhead of calculating multiple path. This refers to number of routing messages and bytes exchanged between routers and the amount of CPU needed to compute paths.

Of the three cost categories, the first is the most performance critical because packet forwarding directly influences the speed in which packets are delivered through the network. Packet forwarding consists of two components: path specification and router forwarding. First, given that every packet must specify its path, path specification must be space efficient to decrease the cost of transmitting and storing packets. Second, router's processing for each packet should be efficient in order to reduce packet forwarding time.

RED HAT 1012
RED HAT V. CAS

The second and third cost categories affect the feasibility of multipath routing. That is, maintaining low router storage and computation overhead is important because if they incur prohibitively high costs, multipath routing cannot be feasibly implemented in real networks. However, these two costs do not directly affect network performance because they do not directly influence packet forwarding speed.

The result of this section's cost analysis shows that capacity removal MPLS and MPDV efficiently implements packet forwarding. Because of the suffix matched forwarding method, data packets in both routing algorithms specify their paths using small, fixed-length path IDs. These IDs, in turn, allow fast path lookups which speed packet forwarding time. This section also shows that both routing storage and route computation costs are reasonably low such that we believe MPLS and MPDV can be feasibly implemented in current routers.

The remainder of this section presents the cost results for each category in the order presented above. The next section addresses the per packet forwarding overhead, followed by router storage and multipath computation overhead.

9.4.1 Per Packet Forwarding Overhead

This subsection examines the per packet forwarding overhead for the capacity removal MPDV and MPLS algorithms. The per packet forwarding overhead refers to 1) the amount of extra information (in bytes) that each packet needs in order to specify the path to its destination, and 2) the additional router processing needed to forward these packets on their specified paths. It is critical to minimize these two costs because they are incurred on every data packet.

Path ID Specification

The suffix matched forwarding method is used in both MPLS and MPDV implementations. This forwarding method guarantees fixed-length path IDs for suffix matched path sets. In particular, the method guarantees the number of bits needed to specify Q paths to a destination is $\lceil \log_2(Q) \rceil$ bits. In MPDV, each router has at most K paths to each destination,

RED HAT 1012
RED HAT V. CAS

where K is the rank. Therefore, the per packet forwarding overhead is $\lceil \log_2(K) \rceil$ bits, and the total cost of path specification is therefore the size of the destination address plus $\lceil \log_2(K) \rceil$.

In MPLS, however, this analysis is not so straightforward. Because routers have to forward packets on non-suffix matched paths, a router may have to specify more than K paths to a destination. For a destination D, the actual number of paths a router has to distinguish to D is number of paths the router calculates to D plus the number of non-suffix matched paths to D that passes through the router (see Figure 8.3). That is, for a destination D, given that a router calculates K' paths and that p' non-suffix matched paths to D pass through the router, the path specification overhead is $\lceil \log_2(K' + p') \rceil$ bits.

This additional per packet forwarding overhead is the price MPLS pays for computing more paths. Notice that the per packet overhead is still $\lceil \log_2(Q) \rceil$ bits, where $Q = K' + p'$.

To make the number of MPLS paths more concrete, in the 170 link topology with $K = 3$, the percentage of non-suffix matched paths is 30%. Therefore on average, a router needs to specify approximately $2K$ paths to a destination (if non-suffix matched percentage is 50%, then a router needs exactly $2K$.). In addition, because each non-suffix matched path potentially needs a forwarding table entry on every router it traverses, the average non-suffix matched path a router has to support is thus $2KL$, where L is the average path length. In the 170 link topology, L is 6. It turns out that the actual number of paths a MPLS router needs to specify to a destination is much less than this upper bound. The next section, which focuses on router storage overheads, re-addresses this issue.

In both MPDV and MPLS implementations, 1 byte is used to specify the path ID. Thus every MPLS router can support up to $2^8 - K$ non-suffix matched paths to each destination. That is, given that $K = 3$, and $L = 6$, the number of non-suffix matched paths a router needs to distinguish is 36 (according to the 2KL formula above), which is much less than $2^8 = 256$. Indeed, in all our simulations, 1 byte was sufficient to uniquely identify all paths to a specific destination.

In summary, the suffix matched forwarding method used in MPLS and MPDV provides

RED HAT 1012
RED HAT V. CAS

efficient guarantees on the size of per packet path IDs. The per packet overhead of path specification for both MPLS and MPDV is fixed-size and small. In MPDV, because it calculates suffix matched paths, the path ID size is precisely $\lceil \log_2(K) \rceil$, where K is the rank. In MPLS, where routers calculate more paths to destinations (via non-suffix matched paths), the cost is higher. However, in practice, a 1 byte path ID is sufficiently large enough for MPLS routers to distinguish all paths to a destination.

Packet Forwarding

Using suffix matched forwarding, the packet forwarding process consists of receiving the packet, looking up the packet's destination address and path ID to find the next-hop address, updating the packet's path ID, and then forwarding the packet to next-hop router/network. Like the per packet forwarding overhead, efficient packet forwarding is critical to network performance because it directly affects the speed at which packets are delivered to their destinations.

The packet forwarding overheads incurred by the suffix matched forwarding method are 1) the additional path ID lookup to find the next-hop address, and 2) updating the packet's path ID.

With respect to forwarding table lookup, the state-of-the art single-path IP forwarding lookup methods hash on destination addresses to retrieve the next-hop address [48, 160]. In a multipath environment, an additional path ID lookup is required. Because of suffix matched forwarding, the path ID is a small integer and therefore can be easily indexed (or hashed) once the destination's forwarding table entry is found. This requires one additional lookup and thus nominally affects routing lookup time. Notice that because hashing is used to determine next-hops, the number of extra paths calculated by a multipath routing algorithm does not have a linear affect on routing lookup time. In addition to hashing speeds, another issue is the size of the forwarding tables. The larger the forwarding table size, the slower or more expensive the memory. The next section shows that MPDV and MPLS forwarding tables are, on average, only a factor of K larger than in single path

RED HAT 1012
RED HAT V. CAS

routing.

The second forwarding overhead is updating a packet's current path ID with the path ID in the forwarding table. This updating process involves writing the new path ID (stored in the router's forwarding table) into the packet's header. Since high speed network switches, such as ones in ATM [130], also perform this procedure on every data unit, we believe that the path ID update procedure can be efficiently implemented in routers as well. Moreover, there are proposed router architectures that explicitly use path ID updates to forward packets [134]. Due to these advances, we believe updating path IDs can be efficiently implemented in multipath routers and will not significantly degrade their packet forwarding efficiency.

9.4.2 Router Storage Overhead

Router storage refers to the amount of memory a router needs in order to support a routing algorithm. In high speed routers, there are two types of router memory, fast memory (e.g. SRAM) and general purpose memory (e.g. DRAM). Since packet forwarding is performance critical, fast memory is typically used to store router forwarding tables, thereby allowing fast forwarding table lookup [48]. General purpose memory is used to store all other information that a routing algorithm uses.

With respect to fast memory, it is important to keep the size of forwarding tables small because larger tables imply slower fast memory access times and/or more expensive (in terms of Dollars) router costs. However, since information stored in the general purpose memory is not performance critical and that the memory is inexpensive, the general storage memory constraint is that it needs to be reasonably low so that multipath routing algorithms can be feasibly implemented in routers.

The experiments measuring router storage overhead is thus divided into forwarding table size and general router storage size. Their measurements are given below.

RED HAT 1012
RED HAT V. CAS

Forwarding Table Size

As proven in Chapter 6, the suffix matched forwarding method guarantees that a router's forwarding table size is proportional to the number of paths the router has to all destinations. Given Q paths to each destination, a router would then need $O(NQ)$ storage for paths to all destinations. This analysis applies directly to capacity removal MPDV because every MPDV router has at most K paths to any destination; thus the storage requirement per MPDV router is $O(KN)$. This storage requirement is only a factor K more than single path DV, which requires $O(N)$ storage.

Capacity removal MPLS storage is not as simple to analyze because the centralized capacity removal algorithm can calculate non-suffix matched paths; therefore a MPLS router may have to store non-suffix matched paths that it did not calculate. In MPLS, a router needs to store, for each destination D, the paths it calculates to D plus the non-suffix matched paths to D that pass through the router. Notice that this sum is also the number of paths a MPLS router has to distinguish. From the analysis in the previous chapter, the per MPLS router forwarding table size is $O(KN + NpL)$, where p is the average number of non-suffix matched paths to a particular destination that pass through a router, and L is the average non-suffix matched path length.

However, in reality, the number of additional forwarding table space is much less because many non-suffix matched paths do not need additional forwarding table entries. Figure 9.9 shows an example.

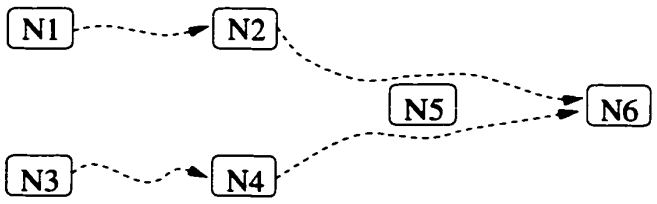


Figure 9.9 : An example of a non-suffix matched path set. Here, N5 does not need to store N1 and N3's non-suffix matched paths.

In Figure 9.9, nodes N1 - N4 computes paths to N6 that passes through N5, but N5 does

RED HAT 1012
RED HAT V. CAS

not calculate the corresponding suffix of the path to N6. The figure shows that there are four non-suffixed matched paths, shown in dotted lines. When router N1 and N3 attempts to find a match for their paths to N6, they will find the correct match in N2 and N4 respectively; therefore, as far as N1 and N3 are concerned, their paths to N6 is suffix matched. In this example, N2 and N4 will fail to find matches in N5. Using the explicit routing method, N5 will add two forwarding table entries (i.e. paths) to make N2 and N4's paths suffix matched. Notice that adding these two paths in N5 automatically makes N1 and N3's paths suffix matched as well.

Due to this property, the actual number of extra forwarding table entries are much smaller than the worse case analysis. Figure 9.10 shows the actual forwarding table storage of MPDV and MPLS routing algorithms on the 100 node, 170 link sparse topology. We show the numbers for a sparse topology because for MPLS capacity removal paths, the number of non-suffix matched paths increases as network connectivity decreases.

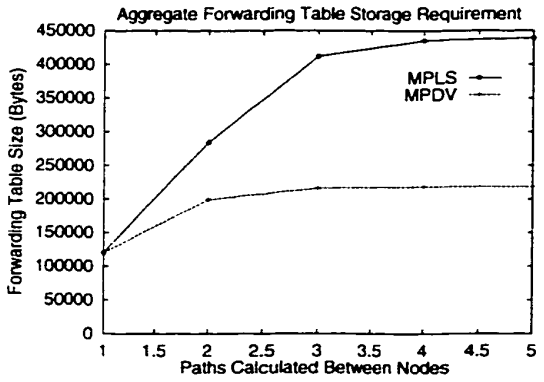


Figure 9.10 : The aggregate forwarding table storage for MPLS and MPDV in a 100 node, 170 link cluster network.

In Figure 9.10, the y-axis shows the aggregate router MPLS and MPDV storage requirement, in bytes, and the x-axis shows the rank of the routing algorithm. Each forwarding table entry is 12 bytes. Not surprisingly, the curves are heavily correlated with the number of paths each algorithm calculates (Figure 9.4). Recall from Figure 9.4, MPLS calculates 22% more paths than MPDV at rank = 3. From Figure 9.10, the extra 22% paths (non-

RED HAT 1012
RED HAT V. CAS

suffix matched paths) that MPLS calculates translate into approximately two times more forwarding table space. Thus, $2NK \sim NK + NpL \Longrightarrow 2K \sim K + pL \Longrightarrow pL \sim K$. That is, experimentally, pL is approximately K.

Since multipath routers compute K paths to every destination, one expects that router forwarding tables need at least K times more storage than their single path counterparts. Indeed, our results show that MPDV adheres to this lower bound while capacity removal MPLS needs more forwarding table space for non-suffix matched paths. Our experiments show that for 100 node 170 and 190 link Internet-like networks, the storage complexity is roughly $2NK$.

To put these asymptotic complexities in perspective, according Figure 9.10, computing 3 paths between nodes requires, *per* router, approximately 4,500 bytes in MPLS and 2,000 bytes in MPDV (divide the y-axis by 100). Given that a 100 node routing domain is quite large, the forwarding table sizes measured here are realistic and applicable to many routing domains. Because the actual sizes of the forwarding tables are fairly small, we believe the amount of fast memory needed to store multipath forwarding tables are not prohibitively costly nor will the performance of the memory be significantly slow when compared to current, single path routing requirements.

General Router Storage Requirement

The second router storage category is general router memory, which stores a routing algorithm's non-performance critical information. Although not performance critical, this storage requirement nevertheless needs to be reasonably low so that multipath routing algorithms can be feasibly implemented. This section experimentally measures the general memory requirements of capacity removal MPDV and MPLS routing algorithms.

In general router memory, the information stored primarily includes the data needed for a routing algorithm to perform path computation. Thus for MPDV, a router stores information of every path it computes, which includes the path's cost, next-hop, and capacity-source route. In MPLS, a router stores the most recent LSPs the router has received from

RED HAT 1012
RED HAT V. CAS

other routers and the source routes of non-suffix matched paths. From the storage analysis in Chapter 8, the per router storage complexity is $O(NKL)$ and $O(NM + NpL)$ for capacity removal MPDV and MPLS respectively. To place these complexities in perspective, Figure 9.11 shows the router aggregate general storage cost for the 170 link network.

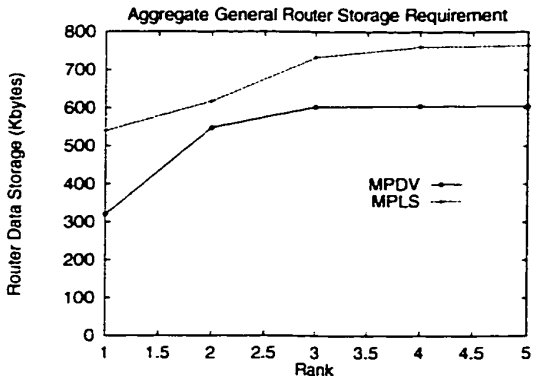


Figure 9.11 : The total general storage cost for MPLS and MPDV in the 100 node, 170 link cluster network.

In Figure 9.11, the x-axis shows the rank of the labeled routing algorithm, and the y-axis shows the aggregate general router storage (in Kbytes) that each algorithm requires. Because the MPDV cost is proportional to the number of paths calculated and the average path length, the cost curve of MPDV has the same shape as Figure 9.4, which shows the number of paths MPDV calculates. The general storage requirement for the 195 link network has the same properties as shown in Figure 9.11.

For MPLS, the general storage cost consist of storing LSPs ($O(NM)$) and the source routes for non-suffix matched paths. Since the number of non-suffix matched paths increases with rank, the MPLS curve in Figure 9.11 increases as rank increases. Although Link State based algorithms typically have to store other information such as the network graph and different data structures used to compute paths, these costs are not shown in here because they are heavily dependent on specific implementations of the path calculation algorithm.

To put these storage requirements in perspective, Figure 9.4 shows that each router

RED HAT 1012
RED HAT V. CAS

needs approximately 6 and 7.3 Kbytes of general memory to store the information needed by MPDV and MPLS to calculate 3 paths per destination. Notice that these memories only measure the information needed by the *routing algorithm*. In practice, a *routing protocol*, which implements routing algorithm, would require additional memory for its internal data structures. However, since protocol costs are need for both single path and multipath routing, these costs do not affect the additional costs incurred by a multipath routing algorithm.

Given that the additional per router general storage requirement of MPDV and MPLS are relatively small (on the order of Kilobytes) in comparison to the amount memory in routers (on the order of Megabytes), we believe that current router's memory capacity can satisfy the additional storage requirements needed by a multipath routing algorithm.

9.4.3 CPU Usage in Path Computation

Router computation cost refers to the CPU cycles needed to compute multiple paths. This cost is not performance critical because path computation is typically performed on a separate processor than ones that forward packets [92]; furthermore, path computations are relatively infrequent compared to packet forwarding. Nevertheless, this cost needs to be low so that path computations can be performed in reasonable time. This subsection analyzes the computational complexities of the MPDV and MPLS capacity removal algorithms.

For the capacity removal MPLS algorithm, the computation analysis is straightforward. The algorithmic complexity of the centralized, naive capacity removal algorithm is $O(KNE * lg(E))$. In addition, since MPLS uses the suffix matched forwarding method, a MPLS router needs to compute, for each destination, paths to that destination from itself as well as from all of the router's neighbors; therefore, the naive computation complexity is $O(MKNE * lg(E)))$, where M is the average number of neighboring routers.

In MPDV, the amount of computation depends on the number of paths each router has to examine and the length of those paths. This is more than the complexity of single path DV which depends only on the number of paths calculated. The reason is that the capacity removal algorithm requires examination of each link in a path. Given that MPDV

RED HAT 1012
RED HAT V. CAS

computes K times more paths and average path length is L, the computation complexity is $O(KL)$ more than SPDV. Given that the average computational complexity of SPDV is $\Theta(N^2M^3(ln(M))^2)$ (Section 8.1), the average complexity of the capacity removal MPDV algorithm is thus $\Theta(KL * N^2M^3(ln(M))^2)$.

We believe optimization such as dynamic programming for MPLS and efficient path encoding for MPDV can reduce the CPU overhead of capacity removal path computation. However, despite the lack of these optimizations, the complexities of the two algorithms are reasonably low; therefore, we believe that they can be feasibly implemented in current routers.

9.4.4 Routing Message Cost

Routing message cost refers to the number of messages needed to propagate information in order to perform route computations. Because our multipath implementations use static metrics, routing message cost is incurred only during topology changes, which occurs relatively infrequently compared to data forwarding. Nevertheless, this cost cannot be prohibitively high because it may affect data forwarding speed (Chapter 3). This subsection analyzes the message cost of MPLS and MPDV algorithms.

As stated in Section 8.1, the average message complexity of capacity removal MPDV is $\Theta(N^2M^3KL(ln(M))^2)$; a factor KL more than the SPDV. This complexity is derived from the product of the size of each MPDVP and the number of MPDVP's transmitted: the size of each MPDVP is $O(NKL)$, and the average number of MPDVP's transmissions is $\Theta(NM^3(ln(M))^2)$.

For MPLS, the routing message overhead comes in two categories. The first category is the messages needed to disseminate topology information. The requirement is the same as the single path LS algorithm, which is $O(NE)$. Notice that compared to SPLS, the MPLS topology messages require an additional capacity specification per link (1 byte); this additional cost does not affect the MPLS message complexity.

The second MPLS message overhead is the cost needed to support non-suffix matched

RED HAT 1012
RED HAT V. CAS

paths. Our implementation of MPLS uses the explicit routing method that source routes non-suffix matched paths. This method does not require any additional computation, but incurs message cost proportional to the number of non-suffix matched paths and their lengths. Thus, the explicit routing message overhead is $O(NpL)$.

However, since the number of suffix matched paths is topology dependent and not all non-suffix matched path need to be source routed (illustrated in Figure 9.9), the analytical bound does not provide a concrete bound. Therefore, we experimentally quantified this cost. Given that message costs increase as network connectivity increases (refer to message complexity analysis), Figure 9.12 shows the routing message cost for MPLS and MPDV in the 195 link topology. The message costs for the 170 link topology exhibit similar characteristics and are not shown here.

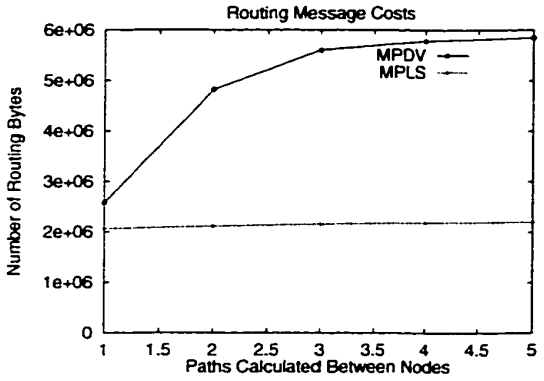


Figure 9.12 : The routing message cost for capacity removal MPLS and MPDV algorithms. The message costs shown are the total number of bytes each routing algorithm transmits for path computation. MPLS uses the explicit routing technique to resolve non-suffix matched paths.

The x-axis in Figure 9.12 denotes the rank of each algorithm, and the y-axis shows the routing message cost (in bytes) needed to compute paths between nodes.

As Figure 9.12 shows, the MPLS message costs increase slightly with the number of paths calculated between nodes. At rank = 1, the MPLS curve shows the cost of LSP broadcasts, which is constant regardless of the rank value. Notice that the MPLS message

RED HAT 1012
RED HAT V. CAS

cost increases slightly with rank. This increase is attributed to the message cost of explicit routing of non-suffix matched paths. The graph shows that the maximum message overhead incurred by source routing (at rank = 5) is approximately 10% of the cost to broadcast LSPs. This means the explicit routing message cost $O(NpL)$ is approximately 10% of the LSP broadcast cost.

For MPDV, notice that the message cost of computing 2 paths per node is approximately the 2 times more than computing one path; however, the cost of computing 3, 4, and 5 paths are only slightly more than computing 2 paths. The reason is that the message cost of MPDV is proportional to the number of paths actually computed. Notice that the curve in Figure 9.4, which shows the number paths MPDV calculates, has the same shape as the MPDV message cost curve in Figure 9.12. This confirms the analysis that MPDV's messaging cost is proportional to the number of paths computed.

In addition to measuring message costs to compute all-pairs paths, another way to gauge message cost is to measure the number of messages needed by a routing algorithm to react to a single topology change (e.g. link failure or recovery). This measure of message cost is given in Figure 9.13. This figure shows the message cost incurred by each routing algorithm to recompute paths when a random link fails and subsequently recovers (i.e. two topology changes).

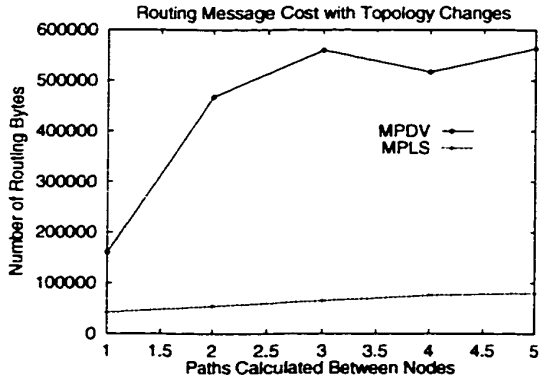


Figure 9.13 : The routing cost of MPDV and MPLS for a single link failure and recovery. The costs show the total number of bytes each routing algorithm transmits to adapt to one link failure and recovery.

RED HAT 1012
RED HAT V. CAS

In Figure 9.13, the x-axis denotes the rank, and y-axis shows the routing message cost when a link fails and subsequently recovers. The MPDV cost curve shows that routing message cost for an isolated topology change has the same general shape as the cost curve in Figure 9.12. However, the proportion of cost difference between 1-rank MPDV and 2-rank MPDV is higher. This is because the 2-rank MPDV has twice the amount of affected paths per topology change, resulting in twice the MPDV message size and two times the amount of updates

The MPLS cost includes topology broadcasts and source routes to establish non-suffix matched path IDs. Again, MPLS costs increase as rank increases. The cost curve in Figure 9.13 shows that the maximum source routing cost (at rank = 5) is approximately the same as the cost for a single MPLS failure and recovery broadcast.

The experiments in this section show that the message costs of both complete and incremental path computations are relatively low. Figures 9.12 and 9.13 show that MPDV costs are proportional to the number of calculated paths and that capacity-source routes contribute approximately L times more routing message cost compared to SPDV. For MPLS, our experiments show that the additional cost of source routing non-suffix matched paths are comparable to the message cost of broadcasting topology changes.

Given that current link speeds are on the order of Megabits to Gigabits per second and that multiple paths are recomputed only when topology changes, the additional multipath message costs, as shown in Figures 9.12 and 9.13, consume a negligible proportion of link bandwidth. Therefore, with the combination of low message overhead, infrequency of route recomputations, and high link bandwidth, we believe the extra messages incurred by MPDV and MPLS will consume a negligible amount of link bandwidth.

9.4.5 Routing Cost Summary

The various costs of executing MPDV and MPLS routing algorithms can be divided into the following five elements: 1) per packet path specification, 2) per packet forwarding time (which depends on forwarding table storage), 3) router general storage requirement, 4)

RED HAT 1012
RED HAT V. CAS

router CPU usage, and 5) routing messages. The first two cost elements are performance critical because they are incurred on every packet and directly affect packet delivery times. The last three categories, although not performance critical, need to be low in order to make the implementation of multipath routing feasible in large-scale networks. The five cost categories are summarized below.

Using the suffix matched forwarding method, the per packet overhead of specifying suffix matched paths is $O(\lceil lg(K) \rceil)$, where K is the number of paths provided a between node pair. This bound holds for all MPDV based algorithms and for MPLS algorithms that compute suffix matched paths. For MPLS that uses non-suffix matched path calculation algorithms, the per packet overhead to specify paths to a destination D is $O(\lceil lg(K + p) \rceil)$, where p is the average number of non-suffix matched paths to D that passes through a router. In our experiments p is very small, $p \leq 5$.

The second cost category, per packet forwarding cost, is also low when using suffix matched forwarding. Because the path IDs are small integers, forwarding packets on multiple path requires only one additional hash lookup, which has minimal impact on router forwarding speed [48, 149, 160]. In addition, the suffix matched forwarding allows compact storage of paths in forwarding tables: the MPDV forwarding table complexity is $O(NK)$, K times more than single path DV. For non-suffix matched MPLS (e.g. capacity removal MPLS), a router's forwarding table complexity is $O(NK + NpL))$. In our experiments, the MPLS storage complexity no more than $2NK$. Thus the suffix matched forwarding method's small, fixed-length integer path IDs not only have low per packet overhead but also allow efficient packet forwarding.

The third cost category, general router storage, is also low. General router storage is not performance critical because the information stored in these memories do not directly affect packet forwarding efficiency. In our implementations, the per router storage for MPDV is $O(NKL)$, KL times more than single path DV, and $O(NM + NpL))$ for MPLS routers. From our measurements, we believe the memory requirements for both MPDV and MPLS can be satisfied by today's routers.

RED HAT 1012
RED HAT V. CAS

Router CPU usage is the fourth cost category. Because more paths are calculated between nodes, the CPU complexities are higher for multipath routing algorithms, compared to their single path counterparts. In MPDV, the CPU complexity usage is proportional to the number of paths examined and their path length ($\Theta(N^2M^3KL(ln(M))^2)$). This is a factor KL more than SPDV. For SPLS, the computation complexity of calculating shortest paths is $O(E * lg(E))$, whereas for MPLS, it is $O(MW)$. Here W is the complexity the multipath calculation algorithm. With the centralized capacity removal algorithm, $W = O(NKE * lg(E))$. Although the actual path computation algorithm has higher complexity, we believe that with the low frequency of path computation, better algorithm design, and advance processor technology, the computation complexity of MPLS will not be a performance bottleneck.

The last cost category is routing message overhead. A routing algorithm's message overhead depends on its path calculation process. For LS based algorithms, message costs are largely independent in the number and type of the paths calculated. This is because LS messages broadcast topology information and are thus independent of path calculation. However, because our capacity removal MPLS implementation source routes non-suffix matched paths, its message complexity is $O(NE + NpL)$. This is compared to SPLS whose message complexity is $O(NE)$. In our simulations, $NE + NpL < 2NE$, for both computing all-pairs paths and for computing paths after a single link failure or recovery.

Unlike LS based algorithms, the message costs of DV based algorithms are closely tied with path calculation because paths are computed through message exchanges. In SPDV, the message overhead is $\Theta(N^2M^3(ln(M))^2)$, and in MPDV, the message overhead is KL more, $\Theta(N^2M^3KL(ln(M))^2)$. MPDV uses a factor of KL more because the capacity removal algorithm needs to examine each path's links, and there are K more paths to examine.

Again, we believe that given current network link speeds and the relative infrequency of path computations, the extra message costs incurred by MPDV and MPLS will consume negligible network bandwidth.

RED HAT 1012
RED HAT V. CAS

Routing Alg.	Per Pkt Cost	Forwarding Tab.	CPU Usage	Rout. Msgs	Exp. Msgs
SPDV	$O(1)$	$O(N)$	$\Theta(N^2M^3(ln(M))^2)$	$\Theta(N^2M^3(ln(M))^2)$	0.72MB
MPDV	$O(lg(K))$	$O(KNL)$	$\Theta(N^2M^3KL(ln(M))^2)$	$\Theta(N^2M^3KL(ln(M))^2)$	5.6MB
SPLS	$O(1)$	$O(N)$	$O(Elg(E))$	$O(NE)$	2.1MB
MPLS-SM	$O(lg(K))$	$O(NK)$	$O(MW)$	$O(NE)$	2.1MB
MPLS-NSM	$O(lg(K+p))$	$O(N(K+p))$	$O(MW)$	$O(NE+NpL)$	2.2MB

Table 9.1 : A summary of the routing costs incurred by different routing algorithms. This table shows the different routing cost categories for the single shortest path DV (SPDV), capacity removal MPDV (MPDV), shortest path Link State (SPLS), suffix matched MPLS (MPLS-SM), and non-suffix matched, capacity removal MPLS (MPLS-NSM) routing algorithms. The last column shows the actual message cost measured for each algorithm on the 100 node 195 link network (for the multipath algorithms, the costs shown are for computing 3 paths). The variables in the table represent the following: N the number of nodes, E the number of network edges, K number of paths calculated between a node pair, p the average number of non-suffix matched paths to a destination that passes through a router, L the average path length, and W the centralized algorithmic complexity of calculating K paths between nodes.

Table 9.1 summarizes the different cost categories. The processing cost for packet forwarding is omitted in the table because it is the same for all five algorithms.

9.5 Experimental Conclusions

This section presents the cost and performance of the capacity removal MPLS and MPDV routing algorithms. The experiments were conducted on two relatively sparse Internet-like, cluster topologies (100 nodes with 195 and 170 links). The results of these experiments show that the implementations of capacity removal MPDV and MPLS routing algorithms provide additional network resources to end-hosts at low routing costs and that end-hosts can effectively use these additional resources to increase their performance.

Network performance is measured in terms of throughput, round-trip latency, and message drop probability. MPTCP is used to measure throughput, and a multipath ping program is used to measure latency and drop probability. With respect to throughput, our simulations show that MPTCP fully utilizes the paths provided by MPDV and MPLS. This

RED HAT 1012
RED HAT V. CAS

observation confirms two important properties of our multipath architecture. First, both capacity removal MPDV and MPLS algorithms compute paths that provide more network resources. Second, MPTCP effectively uses these resources to increase network throughput.

In addition to throughput, our simulations show that multipath networks also allow end-hosts to successfully increase their performance in terms of lower latency and reduced message drop probability. A multipath ping program is used to measure these two performance metrics. Compared to a single path ping program, the simulations show that the multipath ping program achieves 15% lower round-trip delay and incurs approximately 50% less message drops.

The performance sections in this chapter show that multipath networks can provide quality paths and that these paths can be efficiently utilized to increase end-to-end performance.

The second focus of this experimental section is the runtime cost of multipath routing algorithms. The five cost categories are 1) per packet path specification, 2) packet forwarding, 3) router storage, 4) path computation, and 5) routing messages. Of the five costs, the first two are performance critical because they directly influence the speed of packet delivery. Given that MPDV and MPLS both use the suffix matched forwarding method, these two costs are kept at a minimal: fixed-length per packet path IDs, one additional path ID lookup per packet, and small forwarding tables. For the other three cost categories, our experiments show that their overheads are reasonably low, thereby making the implementation of multipath routing possible for large-scale data networks.

The conclusion of this chapter is that multipath routing can provide substantial end-to-end performance improvement, and we believe that the costs incurred by multipath routing algorithms can be satisfied in modern networks.

RED HAT 1012
RED HAT V. CAS

Chapter 10

Related Work

The research presented in this thesis builds on a large body of work. The purpose of this section is to compare and present other work relating to dynamic metric and multipath routing models. Section 10.1 presents related research in dynamic metric routing, and Section 10.2 multipath routing.

10.1 Dynamic Metric Routing

Dynamic Metric Routing Algorithms

The ARPANET was one of the earliest testbeds for dynamic metric routing. Dynamic metric routing was implemented in the early 1980's and used the Link State routing algorithm. This traditionally static metric routing algorithm was modified to compute dynamic metric paths in the following manner: every router continually monitors the costs of its outgoing links. Whenever the difference between a link's current cost and its last advertised cost exceeds a preset threshold, the router then broadcasts an LSP to reflect this cost change. The LSP broadcast initiates a global path recomputation that incorporates the new link cost. A dynamic metric Distance Vector algorithm can be implemented in a similar fashion.

As discussed in Chapter 3, two primary disadvantages of this type of routing algorithms were observed and documented by Khanna and Zinky [94]. First, dynamic metric LS/DV algorithms have the tendency to oscillate traffic from one area of congestion to another. These routing oscillations underutilize network resources and exacerbate network congestion. Second, the amount of routing resources used by both algorithms is unpredictable and hard to control. This problem is worsened because dynamic metric LS/DV's resource con-

RED HAT 1012
RED HAT V. CAS

sumption increases as network traffic increases, thereby competing for network resources at a time when they are most scarce. These two disadvantages limit the wide use of dynamic metric routing in today's Internet.

The hybrid-Scout routing algorithm, to a large extent, overcomes these two limitations (see Chapter 3). Hybrid-Scout reduces route oscillation by independently calculating dynamic metric paths to a selected set of destinations, and its routing costs are easily controlled by Scout generating destinations. The key idea behind the hybrid algorithm is that calculating dynamic metric paths to a selected set of "hot" destinations is sufficient in reducing network congestion, and that such selective calculations can be done efficiently. This idea is made feasible by the high degree of destination locality observed in network traffic today.

Other Measures Of Dynamic Metrics

This thesis develops routing algorithms targeted for large-scale, packet switched wired networks. In these networks, dynamic link metrics are typically used to measure the amount of traffic experienced on a particular link. In other types of networks, however, dynamic link metrics can be used to measure link properties other than traffic.

One such application of dynamic metrics have been applied to wireless, ad hoc mobile networks. In these networks, computers are not physically connected but communicate using wireless channels. The reliability of a connection between neighboring computers depends on factors including distance, the number of interfering signals, etc. Because not all wireless "links" have the same signal strength, Dube et al. in [56] use dynamic link metrics to measure the strength of a wireless connection (measured in terms of signal strength). This link measure is dynamic because a wireless connection's signal strength changes as computers move.

In this context, the routing algorithm provides paths between nodes based not only on path length, but also on the signal strength of each link on that path. In their implementation, packets are routed on paths that use stronger signal links over paths that do not.

RED HAT 1012
RED HAT V. CAS

The routing algorithm used in [56] is a variation of the dynamic metric Distance Vector algorithm where route recomputations are triggered by link cost changes (Chapter 3).

We believe that the Hybrid-Scout algorithm can be applied in this network environment to efficiently calculate strong signal paths. Like wired networks, effectiveness of hybrid-Scout depends on the amount of destination locality in ad hoc networks. However, unlike wired networks which may consist of tens of thousands of nodes, ad hoc networks are typically small, on the order of less than fifty nodes. Given their size, the reduction in routing cost of hybrid-Scout, compared to dynamic metric LS/DV algorithms, will not be as significant as observed in large networks.

Quality of Service Routing

The current Internet provides a best-effort service model: packets are delivered to their destinations without guarantees on the success of packet delivery nor on delivery times. However, with the emergence of applications such as IP telephony and video conferencing, many advocate that traditional best-effort networks should support mechanisms that provide delivery guarantees [17, 29, 40, 86, 162].

These delivery guarantees are generally refer to as *Quality of Service (QoS)* [34, 91, 162, 169]. The QoS of a path is typically specified as a combination the path's delay bound (i.e., packets sent on this path will reach the destination in time less than t) and the path's available bandwidth (i.e., X amount of throughput can be obtained on this path before packet drops). Before transmitting data, an application that requires QoS needs to specify its QoS requirement and make reservations on the path to its destination (i.e. on every router in the path). Once the reservation is made, the application is assured to receive the requested QoS.

Notice, at any given time, the QoS requirement that a router can satisfy depends on the QoS requests the router is currently supporting. That is, the probability a router can satisfy a certain QoS request is dynamic and depends on traffic. Thus, a dynamic link metric can be formed to expresses the probability that a QoS request can be satisfied, and paths can be

RED HAT 1012
RED HAT V. CAS

computed to optimize this metric.

Indeed, there are several proposals to compute paths that consider QoS metrics [9, 11, 72]. Currently, there are two general approaches to implement QoS routing. The first is to use dynamic metric versions of the Link State or Distance Vector algorithms [9, 72] (refer to Chapter 2 for their implementation). The triggering mechanisms allow the two routing algorithms to calculate paths that maximize the probability of satisfying QoS requests. Recall that in datagram networks, routing instabilities tend to occur in dynamic metric routing because path recomputations can potentially shift a large amount of data from one congested area to another. However, this large traffic shift does not occur in QoS routing because once a QoS path is reserved, packets traveling the path are not re-routed, even though the current path between the source and destination may have been recomputed to use a different path. That is, reserved QoS paths are pinned and packets traversing these paths are not affected by subsequent path computations, thereby reducing large traffic shifts and routing instabilities.

We believe the hybrid Scout algorithm be successfully applied to compute QoS paths. However, because routing instabilities and costs are reduced in QoS based LS and DV algorithms, we believe that the cost-performance benefits of hybrid Scout benefits will be less compared to ones observed in congestion-oriented dynamic metric routing.

The second approach to QoS routing is to dynamically calculate a QoS path given a QoS request [10]. In this approach, whenever a router receives a QoS request from a host, the router dynamically computes a path with the highest probability of satisfying the request. Once the path is calculated, the router then forwards the QoS request on that path. To make this approach feasible, the challenge is to reduce the amount of router processing needed for dynamic path computation.

In addition to explicitly considering QoS constraints in the actual routing algorithm, researchers have studied other methods to increase network QoS. One such approach to examine the impact of multipath networks. That is, given a network that offers multiple paths between nodes, how can end-hosts best utilize these paths to better ensure their network

RED HAT 1012
RED HAT V. CAS

QoS to their destinations.

Roa and Batsell in [131] proposed a method which guarantees that end-hosts can transmit their messages with minimal delay, given an accurate view of the network (e.g. up-to-date information of router queues and traffic patterns). Cidon and Rom developed end-host reservation algorithms to increase the probability of reserving QoS request on a multipath network [39]. The different algorithms they developed make different tradeoffs between the probability of satisfying a QoS request with the amount of network resources and time needed to satisfy the request.

Algorithms for Highly Dynamic Networks

Another network environment which requires routing techniques similar to dynamic metric routing is in highly dynamic networks. Highly dynamic networks are characterized by frequent changes in network topology. These networks are typically wireless and highly mobile. Abstractly, dynamic networks can be modeled as a dynamic metric networks where a connected link has cost 1 and a unconnected link has cost infinity, and a link's cost changes between these two values to indicate the link's connectedness (value 1) or disconnectedness (value infinity).

Johnson and Maltz proposed a host-initiated routing algorithm for such networks [87] called Dynamic Source Routing (DSR), which is similar to Scout [33]. The assumption of their work is that the network topology changes so rapidly such that maintaining valid routes to all destinations is infeasible. Thus DSR calculates paths on demand: whenever a node wishes to transfer data to another node, the sending node initiates a path computation to that node. DSR is similar to Scout in that small messages are flooded to calculate paths. However, they differ in several key aspects. First hybrid-Scout maintains paths between all nodes; in an environment such as the Internet, DSR's on-demand path calculation is prohibitively expensive. Second, DSR initiates path calculation from the source, as oppose to the destination. Third, Scout is guaranteed to converge on the shortest path, DSR simply computes a feasible path. Fourth, DSR uses source routing, whereas hybrid-Scout uses

RED HAT 1012
RED HAT V. CAS

hop-by-hop forwarding.

Perkins in [127] proposed a modification to DSR called the Ad Hoc On-Demand Distance Vector (AODV). This routing algorithm is a hybrid of DSR and DV for ad hoc networks. AODV uses DSR for on-demand route discovery of hop-by-hop routes and uses the DV mechanism for invalidating the on-demand paths when link failures are detected. Like the hybrid-Scout algorithm, AODV is an hybrid algorithm. However, unlike hybrid-Scout where both the DV/LS and the Scout component compute routes, the DV component in AODV is used only to invalidate routes discovered by DSR (in a similar way the DV component in hybrid-Scout invalidates Scout paths). Again, the AODV algorithm cannot be directly applied to large-scale datagram networks because on-demand route discovery is too expensive.

10.2 Multipath Routing

This section surveys related work on multipath routing. The discussion is organized as follows: the first section describes other designs and/or implementations of multipath networks in comparison with the one proposed in this thesis. Section 10.2.2 discusses other work specifically related to algorithms that calculate multiple paths, and the last section surveys different proposed path forwarding methods.

10.2.1 Multipath Networks and Architectures

IBM SNA

The IBM Systems Network Architecture (SNA) network is a wide area network first deployed in 1974 [70]. This network provides multiple paths between nodes because its designers thought that the traffic generated by routing algorithms that adapt to network failures/recoveries would incur prohibitively high routing costs* [6]. Therefore to account for network failures, multiple paths between nodes are centrally pre-computed so that if one

*This is known to be untrue today.

RED HAT 1012
RED HAT V. CAS

path fails, other paths would still be operational. SNA also has a notion of multi-service and multi-option paths: the service type are interactive and batch which are used to dictate packet priority.

Conceptually, the SNA multipath model is very different from the one developed in this thesis. First, the purpose of SNA's multiple paths is to provide fault-tolerance without using adaptive routing; whereas in our model, fault-tolerance is provided by the adaptiveness of the routing algorithm, and multiple paths are used solely to increase network performance and services. Second, path services in SNA are predefined. In contrast, the proposed multipath model is designed to accommodate future path services that have yet to be defined.

In practice, SNA also differs significantly from our proposed multipath network. First, SNA's multiple paths are *predefined* and *centrally pre-computed*. That is, path computation is done centrally off-line where routing tables are distributed after this computation is complete. Since SNA routing is not distributed and not adaptive, it cannot be practically deployed in wide-area networks of today's scale. Secondly, the SNA network restricts an application to use only one path at a time, and the path is specified at connection time. On the other hand, our multipath routing model allows full application freedom on the usage of multiple paths.

Scalable Routing Architecture

Estrin et al. proposed a more recent multipath routing model targeted for today's large-scale Internet environment called the Scalable Inter-domain Routing Architecture (SRA) [59]. This proposed multipath routing scheme supports two types of paths. The first is the static metric single shortest paths calculated by a traditional DV routing algorithm. Packets traveling on these paths are hop-by-hop forwarded as in the current Internet. These "generic" paths are calculated by routers, and it is expected that the majority of network packets will use these generic paths. The second type of path called Type Of Service (TOS) path, is calculated by end-hosts. Packets sent on TOS paths are forwarding using IP loose source routing [57]. Calculation of these paths requires end-hosts to first collect the network topology

RED HAT 1012
RED HAT V. CAS

information from routers and then individually calculate the desired path(s). Since there are no restrictions on path calculation, applications in SRA can send packets on arbitrary paths.

Although SRA and the multipath model considered in this thesis are targeted for the same network environment, they are fundamentally very different. In SRA, packets traveling on non-shortest paths (i.e. on TOS paths) are expected to be few in comparison to packets on generic paths. This assumption is reflected in the difficulty in which TOS paths are calculated and the inefficiency of the packet forwarding method. Whereas, in the multipath model proposed, all paths are calculated with the same efficiency and packets are forwarded the same way irrespective of the path they travel. In short, our multipath model views packets traveling non-generic paths as the general case, whereas in SRA, they are viewed as the exception. This perspective difference is evident in each model's path calculation and packet forwarding methods.

Detour

As mentioned in Chapter 2, Internet routing is decomposed into routing domains for scalability reasons. Routing domains are typically divided by domain ownership, as opposed to domain geographies. This causes routing inefficiency because the path between two geographically close nodes may actually traverse domains that are geographically very far from the source and destination domains. In addition, inter-domain routing algorithms calculate paths based on minimizing the number of domains traversed. This policy also gives rise to sub-optimal routing because the time needed for a packet to traverse a domain depends on the size of the domain, which is highly variable. The result of domain decompositions and inter-domain routing policy frequently causes routing anomalies because minimizing the number of domains in a path does not necessarily minimize path delay. For example, sending a packet "directly" from domain A to B may take longer than sending the packet "indirectly" from A to C and then to B.

The Detour project aims to overcome such routing efficiencies caused by these two

RED HAT 1012
RED HAT V. CAS

factors [8,136]. Detour's approach is to establish a set of "smart" routers that communicate with each other to setup a virtual network. The nodes of the virtual network are Detour routers, and a virtual link connecting two Detour routers is the path in the physical network that connects the two routers. Packets sent by one Detour router are tunneled [134] to its neighboring router. The cost of each virtual link is computed dynamically by sampling the link's delay. In the previous example, Detour hopes that its routers will automatically reroute packets sent from A to B through C using this virtual network.

Detour also plans to support "multipath routing": if Detour routers detect that there are multiple, virtual paths to a destination, the routers will distribute data among the multiple paths. Notice that this type of multipath routing differs from the model proposed in this thesis because end-hosts in Detour do not have control over which path to use. As stated in Chapter 4, the absence of end-host control limits the performance gains of multipath networks.

In short, Detour aims to address inter-domain routing inefficiencies by establishing a virtual network that delivers packets based on actual observed path latencies (i.e. not based on the number of domains traversed). Detour also plans to support a form of multipath routing where end-hosts do *not* have control over which path to send their packets.

We believe the Detour architecture may be used to incrementally deploy the multipath routing model described in this thesis. Using the virtual network, Detour-like routers can implement end-host controlled multipath routing. Details of this approach are given in Chapter 11.

Specialized Networks

Another network environment where using multiple paths have been proposed and implemented is in short-hauled networks [50,106,166]. Short-hauled networks are characterized by tightly coupled network elements and are usually implemented in multiprocessor networks or switches. Examples of short-hauled networks are Banyan networks [140], Omega networks, mesh topology networks [143], cube topology networks [37,55,105,143], gen-

RED HAT 1012
RED HAT V. CAS

eral multistage networks [49, 104], and internal ATM switching elements [76, 88, 145, 155, 164].

The goals for providing multiple paths in a short-hauled network are to decrease switch blocking probability (e.g. switch overloading) and to increase fault tolerance [132, 161]. In addition to scale, the main conceptual difference between a short-hauled network and a wide area network is that paths between nodes within short-hauled networks are hardwired. That is, paths between nodes are determined at network design/construction time, as oppose to one that is dynamically determined and adapts to component failures.

Hardwiring paths is possible in short-hauled networks because node addresses are chosen so that the network can determine the path to a node by inspecting the node's address. In this well defined environment, short-hauled network designers are able to assign, *a priori*, multiple paths between nodes.

Unfortunately, the methods employed in these networks are not directly applicable to wide area networks. The three main reasons are 1) wide area networks are not tightly coupled, 2) the wide area network environment is dynamic (i.e. node and links may fail), and 3) it is not possible to pre-assign paths to nodes based on node address.

10.2.2 Multipath Calculation Algorithms

This section presents related work on multipath calculation algorithms. Unlike the multipath networks presented in the last section, the algorithms described here are concerned with path computation and are generally not designed for a distributed network environment. Nevertheless, these algorithms are important because they provide the basis on which to build distributed multipath calculation algorithms.

Introductory algorithm text books such as [19,42,108] provide basic path computation algorithms. For more advanced graph algorithms, refer to [5].

RED HAT 1012
RED HAT V. CAS

Disjoint Paths

The most obvious multipath calculation criteria is to calculate multiple disjoint paths. There are two types of path disjointness, node and link disjointness. A set of paths between two nodes is node (link) disjoint if and only if all paths in the set do not have any node (link) in common [152]. Several algorithms are surveyed below.

The algorithm presented in [118] computes pair disjoint paths between nodes such that the total costs of the paths are minimal. Later work by [36] extended the algorithm to compute K disjoint paths with guaranteed total minimal path cost. Notice that both of these algorithms do not guarantee that the least cost path is in the computed set.

Sidhu et al. developed a distributed multipath calculation algorithm that calculates node disjoint path and guarantees that the shortest path is in the calculated set [146]. This algorithm calculates path from node X to Y by considering the relative position of Y on the shortest path tree rooted at X. Alternate paths are calculated by finding cross edges that connect Y to X. The two major drawbacks of this algorithm are that 1) the algorithm may diverge if the network topology changes during path computation, and 2) the worst case message complexity is exponential.

A variant of disjoint paths is initial link disjoint paths. Initial link disjoint paths, described in [156], is a centralized, dynamic programming algorithm that calculates K paths to every node such that the paths to a particular node has distinct initial links. The algorithm is based on Dijkstra's shortest path algorithm [54].

Shortest K Paths

Another criteria for calculating multiple paths is to calculate unconstrained K shortest paths. The objective of this criteria is to find K distinct paths between two nodes such that they have the minimum sum compared to all other K possible distinct paths. Many algorithms have been developed to compute these paths; refer to [103,141,165] for different approaches. The computational lower bound for computing unconstrained K shortest path is $O(m + n\log(n) + kn)$, where m is the number of edges, n number of nodes, and k the

RED HAT 1012
RED HAT V. CAS

number of path to calculate between nodes [58].

Multi-commodity Flows

Perhaps the most well study of all multipath calculation algorithms are ones that compute multi-commodity flow paths. The constraint of a multi-commodity flow problem is to minimize the total cost of transmitting multiple resources to and from different nodes. This optimization is constrained by the capacities of each link for each resource and the costs of using a link [5,63]. For more references on this subject, refer to [13, 14, 89].

A variant of multi-commodity is maximum flow. A maximum flow algorithm computes paths between two nodes such that the aggregate capacity of the calculated paths is maximized between the two nodes. These paths can be calculated efficiently and are applicable to routing algorithms that wish to provide high capacity paths between nodes [5]. The capacity removal algorithm presented Chapter 5 also computes paths to increase capacity. But unlike maximum flow algorithms, the capacity removal algorithm considers path length and the number of paths calculated.

10.2.3 Multipath Forwarding Methods

As Chapter 4 showed, forwarding messages on multiple paths is significantly harder than on single shortest paths because paths in a multipath network are not guaranteed to be suffix matched. For paths that are suffix matched, Chapter 6 presented an efficient forwarding method that require fixed-length per packet path ID and router storage linear to the number of paths provided. This section surveys other path forwarding methods.

Source Routing

Source routing is a forwarding technique in which the sender of a message determines the path a message is to travel and tags the message with that path. That is, a message contains an explicit list of nodes the packet is to traverse. Upon receiving such a message, a router simply forwards the message to the neighbor specified in the message's source route.

RED HAT 1012
RED HAT V. CAS

This method is very flexible because messages can be forwarding on arbitrarily paths. However, source routing has two major drawbacks. First, with each message carrying an explicit path, source routing's message overhead is variable length and high for large networks and long paths. Second, for each router, the storage requirement for each path is $O(LP)$, where L is the average path length, and P is the size to encode each element in a path. This requirement is compared to $O(1)$ in conventional single path routing algorithms and the proposed suffix matched forwarding method.

Label Swapping

Label swapping performs path forwarding by using local path identifiers [139]. In label swapping, a path $X = (x_0, \ldots, x_n)$ is implemented by nodes x_i and x_{i+1} sharing a local path identifier (LPID) such that whenever x_i receives a message with $lpid_{i-1}$, it forwards the message to x_{i+1}, replacing the LPID with $lpid_i$, $0 < i < n$. Examples of label swapping are the suffix matched forwarding and ATM's virtual circuit forwarding methods.

The advantage of label swapping is that the message overhead and the per path router memory overhead are constant. However, the difficulty in label swapping is the establishment of LPIDs such that they globally compose the calculated paths. For example, ATM requires a special pre-connection packet that traverses the intended path to establish LPIDs. Notice that in the suffix matched forwarding method, however, LPIDs are implicitly made consistent by exploiting the suffix matched property and therefore do not need pre-connection packets.

The label swapping technique has also been applied to make source routing more efficient [85]. For example, the solution adopted in Chapter 6 uses source routing with label swapping to establish LPIDs for non-suffix matched paths. Using this optimization, a path is source routed only once, and label swapping is used subsequently. This optimization greatly amortizes the initial source routing overhead.

RED HAT 1012
RED HAT V. CAS

Compact Addressing

Source routing lists nodes on a path where nodes are identified by a node identifier, which can be very large (in number of bits). For example, a node on the Internet has an ID that is 32 bits long [30] and will increase to 128 bits with IPv6 [26]. Compact addressing reduces the amount of bits needed to enumerate a path by observing that nodes only forward messages to their neighbors. Therefore, if a node has a local name for each neighbor, say $w_1, \ldots, w_M$ for neighbors 1 to M, then a node only needs $\lceil \log(M) \rceil$ bits to forward messages to its correct neighbor [67]. Assume that a node has at most M neighbors, then to encode a source route of length l takes $l * \lceil \log(M) \rceil$ bits. Since M is typically not very large (< 20), this optimization can substantially reduce the overhead of source routing packets.

RED HAT 1012
RED HAT V. CAS

Chapter 11

Conclusions and Future Directions

The contributions of this dissertation are the methods developed to implement dynamic metric and multipath routing models, and the experimental analysis which demonstrates that the two models can be implemented efficiently and significant performance gains obtained. The two routing models, compared to the current Internet's static metric single path model, better utilize network resources to provide higher network performance.

For the dynamic metric routing model, this thesis develops a novel routing algorithm, hybrid-Scout, that increases network performance by exploiting the destination locality observed in network traffic. For multipath routing, this thesis develops not only efficient multipath routing algorithms, but also a transport protocol that effectively uses multiple paths.

This chapter concludes the research presented in this dissertation and outlines potential future research directions.

11.1 Conclusions

Dynamic Metric Routing

This dissertation began with the development of a dynamic metric single path routing algorithm, the hybrid-Scout. Traditional dynamic metric routing algorithms based on Link State and Distance Vector suffer two primary drawbacks: the tendencies for routing oscillations and uncontrolled routing costs. The hybrid-Scout algorithm addresses these two problems through selective, time staggered path computations that are controlled by destinations.

The algorithm is selective because it calculates dynamic metric paths only to certain

RED HAT 1012
RED HAT V. CAS

"hot" destinations – analysis of Internet traffic locality indicates that a few destinations receive the majority of network traffic (the top 1% destinations receive 60% of network traffic); thus calculating dynamic metric paths to these destinations significantly impacts the performance of the entire network. Moreover, the dynamic metric paths are calculated in a time staggered manner. These two properties of the hybrid-Scout algorithm, selective and time staggered route computation, reduce route oscillations because not all paths are calculated at the same time.

In addition to increasing route stability, the routing costs incurred by hybrid-Scout are also predictable. In traditional dynamic metric routing algorithms, routing traffic are triggered by network traffic, making them unpredictable and traffic dependent. In hybrid-Scout, however, routing traffic are controlled by the destinations whose paths are dynamically calculated. Therefore, routing costs are easily controlled by bounding the routing costs of these destination.

To validate the hybrid-Scout algorithm, extensive simulations were conducted to measure its effectiveness and efficiency and to determine the conditions under which hybrid-Scout is better than dynamic metric LS and DV. Simulation results show that, for an Internet-like topology, 1) hybrid-Scout is effective at rerouting network congestion, under the condition that at least 50% traffic are destined to selected destinations, and 2) hybrid-Scout is more efficient (by 1 to 2 orders of magnitude in routing cost) than both dynamic metric LS and DV if less than 10% of nodes generate Scouts. Since these conditions are present in today's Internet, we conclude that hybrid-Scout, if implemented, can improve Internet performance.

Multipath Routing

The second main contribution of this dissertation is the development of methods that implement and utilize multipath networks. Multipath networks are networks that provide multiple paths between nodes at the same time, and transmitting nodes can select which path(s) to use. With respect to multipath routing, the contributions of this dissertation are

RED HAT 1012
RED HAT V. CAS

the design and evaluation of algorithms that make multipath routing feasible for large scale data networks.

To this end, solutions to solve two main multipath problems are developed: 1) routing algorithms that efficiently support multiple paths between nodes, and 2) an end-host protocol that effectively uses multiple paths to *actually* increase throughput performance.

The first problem, supporting multiple paths efficiently, is primarily concerned with the cost of specifying and forwarding packets on their intended paths. While trivial in single path routing, this process is difficult in a multipath context because there are more than one path to a destination. To solve this problem, this dissertation develops a novel forwarding method, the suffix-matched forwarding method, that uses fixed-length per packet path IDs and requires router storage linear to the number of paths calculated. The method's low overhead is achieved by exploiting path sets with the suffix matched property. To demonstrate the applicability of suffix matched forwarding, this thesis extends the method to multipath versions of the LS and DV routing algorithms.

The second multipath contribution of this thesis is the development of an end-host protocol that effectively uses multiple paths to increase end-to-end and network throughput. This thesis develops a multipath transport protocol, MPTCP, that provides a reliable bit-stream service and effectively uses multiple paths to maximize throughput. Extensive simulations show that MPTCP is able to increase throughput proportional to the amount of available network resources. Furthermore, MPTCP sustains high network performance even during very high levels of network utilization.

In addition to performance, this thesis also measured the costs incurred by two multipath routing algorithms, MPDV and MPLS, based on DV and LS routing algorithms respectively. Because both algorithms use suffix matched forwarding, the performance critical operation of packet forwarding uses fixed-sized, per packet path IDs and adds only one additional lookup in router's packet forwarding procedure. With respect to non-performance critical costs, simulations show that the overhead of router CPU usage, router message cost, and router storage requirements are reasonably low so that we believe the two routing

RED HAT 1012
RED HAT V. CAS

algorithms can be feasibly implemented in today's network routers.

11.2 Future Directions

Two general directions of research can follow from the work presented in this dissertation. The first is the improvement of the hybrid-Scout dynamic metric routing algorithm, and the second is the further development of the multipath routing model.

11.2.1 Hybrid-Scout

The efficiency of the hybrid-Scout algorithm depends on a network's destination locality: experiments show that if at most 10% of the destinations (hot destinations) generate at least 50% of the network traffic, hybrid-Scout is more efficient and effective than traditional dynamic metric routing algorithms. This thesis assumes that the hot destinations are known and that each destination generates Scouts at a rate appropriate for the destination's received traffic. A future research direction for the hybrid-Scout algorithm is to remove the two assumptions: develop a mechanism that determines the set of hot destinations and the Scout generation rate for each hot destination.

To accomplish this, a distributed traffic monitoring protocol can be installed in network routers. The protocol gathers traffic statistics to determine which destination networks are "hot". Once this determination is made, routers directly connected to these destinations can then generate Scouts on each of those destination's behalf. In addition, since routers have traffic statistics of all hot destinations, the appropriate Scout generation rate can be derived for each destination.

The development of mechanisms to determine hot destinations and their Scout generation rate will make hybrid-Scout easier to deploy. With these mechanisms, hybrid-Scout will not require manual configuration of Scout generating destinations and their rates.

RED HAT 1012
RED HAT V. CAS

11.2.2 Multipath Routing

This dissertation laid the ground work for supporting multipath routing in large-scale networks. Because applications of multipath routing have largely been unexplored, there are many open research issues. Among others, a list of avenues worth pursuing are 1) the support for different path services 2) the investigation of cooperative multipath transport protocols, and 3) the incremental deployment of multipath routing in an Internet-like network. Each of these possibilities are described in detail below.

Supporting Different Path Services

This dissertation examines two path services, low delay and high throughput services. These two services are chosen because of their prevalent use. However, with increasing diversity of network applications and demands, the need for paths with other services has increased. Examples of these new services are QoS, security, real-time, and network pricing. Possible research directions are to develop multipath methods to efficiently support these path services. To give an example of the issues involved in supporting each of the mentioned services, an outline of some of the problems involved in supporting the QoS path service is given below.

To provide QoS path services, routers need to enforce queuing disciplines to provide different types of delivery guarantees [95]. Example QoS services are guaranteed, priority, and bounded services [11, 43]. Current proposals to support QoS have been primarily focused on the single path routing model. In a multipath setting, if an end-host has can make QoS requests on multiple paths to a destination, one expects that QoS requests can be more easily satisfied. Doing this raises two research issues: how should multiple paths be calculated in order to consider router QoS guarantees? And how end-host applications should best use these paths to most efficiently obtain their QoS requirements?

The first issue is concerned with path calculation: designing path calculation algorithms such that the paths calculated between nodes provide the maximum likelihood of satisfying end-host QoS requests. As stated in Chapter 5, end-hosts can benefit from a multipath

RED HAT 1012
RED HAT V. CAS

network only if the paths provided offers higher performance than in a single path environment. The second issue relates to designing protocols that efficiently use network resources to provide QoS guarantees. For example, a QoS protocol should have the ability to combine QoS guarantees of individual paths to provide an overall QoS for an application. Protocols to accomplish this task need more sophisticated path management strategies than the scheme MPTCP uses for dynamic load balancing.

Transport Protocols

Another possible area of research is to design a multipath transport protocol that uses multiple paths in a cooperative manner. Recall that MPTCP uses separate TCP connections on each available path, and each path is solely responsible for transmitting its own packets. In a cooperative protocol, a more flexible form of load balancing can be used and many redundancies may be reduced.

For example, a cooperative protocol can load balance lost segments from one connection to another. As mentioned in Chapter 7, this reduces receiver buffering which can increase protocol throughput. A cooperative protocol can also reduce redundancies. For example, receivers can send one ACK to acknowledge data received on several connections, and the combined ACK should travel the fastest path back to the sender.

Multipath Incremental Deployment

The algorithms and techniques developed in this dissertation are designed and evaluated under realistic network assumptions. Thus, the natural research question is how to deployed the proposed method in an actual large-scale, wide area network.

In practice, deploying algorithms in wide area networks is done in an incremental fashion. The incremental process is necessary because of the network sizes and because different parts of these networks are often under different administrative controls. These two factors make simultaneous routing algorithm updates of every router in a large-scale network impractical. Thus, a practical research subject is developing a method to incrementally

RED HAT 1012
RED HAT V. CAS

deploy the multipath routing algorithms developed here.

As mentioned in Section 10.2.1, a Detour-like architecture can be used to incrementally deploy the proposed multipath model [47,115,136]. Using this approach, a router with multipath capabilities establishes a virtual multipath network with other multipath-routers. In this virtual network, the nodes are multipath-routers, and the links are established between these routers using IP tunneling [134]. With this architecture, only end-hosts whose local router is a multipath router can use the multipath network because the local router is responsible for translating host multipath IDs into suffix matched path IDs. In incremental deployment, the connectivity of the virtual network increases as more multipath routers are placed in the network, resulting in higher performance of the multipath network.

RED HAT 1012
RED HAT V. CAS

Appendix A

The Scout Routing Algorithm

This chapter describes the Scout routing algorithm. Scout is a destination initiated, selective shortest-path computation algorithm. The key idea underlying Scout is that a destination periodically floods small, fixed-sized messages through the network which explore available paths back to the destination*. These exploratory messages, called *Scouts*, are of a fixed, small size and can be readily piggy-backed onto data packets on a hop-by-hop basis, which largely defrays their cost to the network.

The remainder of this chapter is organized as follows. The next section describes the Scout routing algorithm, followed by Section A.2 which proves its correctness. Concluding remarks are given in Section A.3.

A.1 Flood Based Route Discovery

To facilitate the presentation, the Scout algorithm is presented by initially assuming idealized network characteristics. These assumptions are then progressively removed and additions provided to deal with the relaxed assumptions.

Ideal Network: To feature the basic concepts of the algorithm, assume

- Static network topology where all links have unit cost.
- Message delay along each link is equal.

We define the *shortest-path tree* rooted at node R as a tree which connects all reachable nodes to R with the least cost.

---

*In general a destination could represent a single host, a subnetwork, or an entire routing domain.

RED HAT 1012
RED HAT V. CAS

The basic mechanism of route discovery is through message flooding. In the Scout algorithm, each node in the network periodically sends a Scout message to all its neighboring nodes. Let R be the node initiating the Scout message. The period between two consecutive floodings of Scout messages from R is called the *broadcast interval* (BI). A Scout $[R, C_R]$ contains the originating node's address, R, and the cost to reach R, C_R. Initially C_R is zero. When a node P receives a Scout message $[R, C_R]$ from its neighbor Q, P first modifies the Scout's cost C_R to include the cost of sending a message from P to Q, $C'_R = C_R + Cost(P \rightarrow Q)$. C'_R represents the cost a message will take if it is sent by P via Q to R. Under the two assumptions above, if P has already seen a Scout from R in the current broadcast interval, P already has a route to R with cost at least C'_R. In this case, P remembers this alternate path but does not forward the Scout. Otherwise $[R,C_R]$ is the first Scout P has seen in this broadcast interval, then P forwards $[R, C'_R]$ to all its neighbors except Q. Since nodes only forward R's Scout at most once in each BI, the flooding of Scout messages terminates after every node has forwarded a Scout message of the form $[R, C_R]$. This means the total number of Scouts exchanged in each broadcast interval for a single node is $O(L)$, where L is the number of links in the network.

After the termination of R's BI, every node in the network knows the minimum path to R. This knowledge is represented in the form of a next-hop address: node P the minimal path to R if P knows the minimal cost to R and which of its adjacent nodes is on the minimal path. In the Scout algorithm, node P knows the shortest cost to R and the shortest next-hop to R when P receives the first Scout message in a BI. Nodes decide the shortest next-hop to R as the neighbor from whom they received the least cost Scout in the current BI. The forwarding tree constructed is a *sink* tree rooted at R. Since nodes only advertise (or forward) the least cost route to R, the Scout algorithm guarantees that all nodes will know the minimum path to R when flooding terminates. All pairs shortest path is computed once every node in the network floods its Scout message; thus the worst case time complexity of the Scout algorithm is $O(NL)$ messages, where N is the number of nodes in the network.

RED HAT 1012
RED HAT V. CAS

To disambiguate Scout messages from one broadcast interval with another, assume for the time being that before R floods, all messages of R's previous flood are terminated. This requires the BI to be at least the time it takes for a message to traverse the longest path in the network. This assumption is removed in the general algorithm.

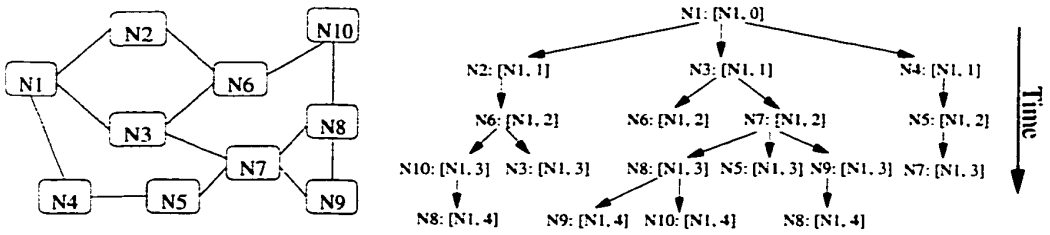


Figure A.1 : Example network topology and the first round broadcast tree from N1. All link costs in the network are 1.

Figure A.1 shows the *broadcast tree* for Scout message flooding initiated by N1 in the above network under our idealized assumptions. A broadcast tree represents an event ordering of sending and receiving messages.

The figure illustrates a sequence of events in which Scouts are sent and received after N1 initiates a broadcast interval. A node Ni: [Nx, k] in the tree represents an event in which node Ni receives a Scout [Nx, $C'_{\mathrm{N}x}$] . Edges show Scout message transmissions and time flows downward in the figure. For events on the same level of the tree, we adopt the convention that events on the left happen earlier than events on the right. For example, event N6: [N1,2], the child of N2: [N1,1] happens before N6: [N1,2], the child of N3: [N1,1]. That is, node N6 receives a Scout from N2 before receiving one from N3, therefore as the figure shows, N6: [N1,2] child of N2: [N1,1] forwards the Scout while the other does not. Leafs of the tree correspond to events which occurred on nodes who have already received a Scout from N1. As Figure A.1 shows, the shortest path is computed when flooding terminates.

Non-uniform link cost and delay: The Scout algorithm described above relies heavily on the assumption that the first Scout message of a BI is the one with minimal cost. Next, this assumption is removed by allowing non-unit link costs and non-uniform (but bounded)

RED HAT 1012
RED HAT V. CAS

link delays. For now, the assumption that all Scouts of R are terminated before R's next flood and that the network topology is static is preserved.

We define two terms. Node P's *designated neighbor* to R in the broadcast interval i is defined as the neighbor that gave P the least cost Scout to R in broadcast interval $i-1$. P's *upstream node* to R in broadcast interval i is the neighbor which provided the Scout that P forwarded to its neighbors (after modifying its cost).

As with the previous algorithm, every node R periodically floods its Scout message $[R, C_R]$. Upon receiving $[R, C_R]$ from a neighbor, P computes C'_R as before. In the first broadcast interval, immediately after receiving the first Scout message from R, node P forwards $[R,C'_R]$ to all neighbors except R. Node P might receive more of R's Scouts in the same BI, indicating different paths and path costs to R. P remembers these Scouts and adjusts its data forwarding table to reflect the new paths, but P does not forward the Scout messages. In the next broadcast interval, P waits to receive a Scout message from its designated neighbor before flooding. When P receives the Scout from this designated neighbor, P computes C'_R and examines all other Scout messages it has received in the current BI (including the one from its designated neighbor), to find the Scout with the least cost C''_R to R[†]. P forwards $[R, C''_R]$ to all neighbors except the neighbor from which it received the Scout with the least cost. This neighbor is now P's upstream node for this BI.

As an example, consider the network in Figure A.1 with the cost of edge (N1, N2) and (N1, N3) set to 5 and all other edge weights being 1. In the first broadcast interval of N1, the broadcast tree is exactly the same as in Figure A.1, except the cost of the N2 and N3 subtree is increased by 4. Figure A.2 shows the broadcast tree for the second broadcast interval of N1.

Notice in Figure A.2, N7 does not forward the first Scout it receives (one from N3) but waits for N5 (its designated neighbor) and then forwards the Scout with the least cost. The shortest-path tree is computed in the third broadcast interval when N3, N8 and N9 wait for

[†]In general, with a dynamic network, the least cost Scout may not come from the designated neighbor, since link costs may change between broadcast intervals.

RED HAT 1012
RED HAT V. CAS

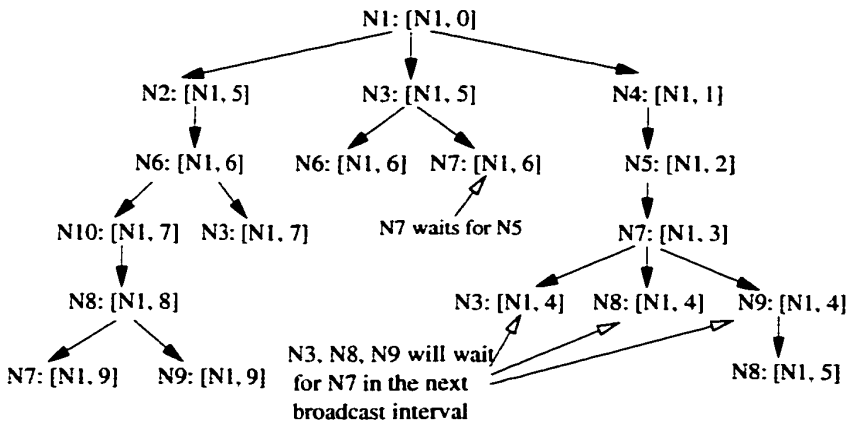


Figure A.2 : Broadcast tree after the second round of flooding from N1 in the network of Figure 1 with unequal link costs.

N7 and forward the least cost Scouts to N6 and N10.

Under these network assumptions, the shortest path computation to any node R is bounded by K broadcast intervals, where K is the depth of the shortest-path tree rooted at R. In the example above, the depth of the shortest-path tree is 5 and the Scout algorithm converged in 3 broadcast intervals. It is worth mentioning that although the shortest-path computation is bounded by K broadcast intervals, all nodes know *a* path to R after R's first broadcast interval. The quality of the path progressively improves with additional broadcast intervals.

Unrestricted Network: This is the generalized network where nodes/links can fail and recover and link costs can change.[‡]. The static network assumption guarantees that if node P receives a Scout message in the previous broadcast interval from neighbor Q, P will also receive a Scout message from Q in the current broadcast interval. If links are allowed to fail and recover, P might never receive another message from Q. To resolve this problem, Scout is modified by requiring P to flood the first Scout message from R if P did not flood any of R's Scout messages in the previous BI. In other words, if P waited for a Scout from

[‡]Notice that convergence of a shortest path computation applies only when the network is not changing for some period.

RED HAT 1012
RED HAT V. CAS

its designated neighbor Q in the previous BI but never received a Scout, instead of waiting for Q, or any other neighbor in the current round, P immediately floods the first Scout it sees in the current broadcast interval and recalculates its designated neighbor.

P is not allowed to wait for a neighbor in the current BI, in particular, the designated neighbor, because if there are multiple failures, waiting for the best information might cause cascading waits. This decision was motivated by the observation that propagating more recent, perhaps sub-optimal, information is more useful than trying to wait for the best information which entails the risk of not propagating any information at all.

To handle overlapping broadcast intervals from the same source, Scout messages are simply tagged with a sequence number indicating the current broadcasting interval. The sequence number ensures that the algorithm only makes routing decisions based on the information from the most recent broadcasting interval. Figure A.3 summarizes the general Scout algorithm.

1. Destinations periodically generate a Scout with an increasing sequence number.
2. On receiving a Scout, the router discards the Scout if
   the Scout sequence number is not current
   OR the Scout advertises a path to the current node.
3. The router adds the cost of the incoming link to the cost of the Scout and
   A. If the router has not forwarded a Scout (from the same source) in the last BI, flood the Scout.
   B. Else if Scout is from the designated neighbor, forward the least cost Scout (from the same source) received in the current BI.
   C. Else store the Scout.
4. Update forwarding table to reflect the shortest path.

Figure A.3 : Scout Algorithm Summary for Unrestricted Networks

Scout and Distance Vector Routing Algorithms

Both the Scout and the Distance Vector routing algorithms compute shortest paths between nodes by exchanging messages that convey path cost information. In Scout, if a router receives a Scout message from neighbor Q originated from node A with cost C, the router knows that it can reach A via Q with cost C. Similarly, in DV, if a router receives a DV

RED HAT 1012
RED HAT V. CAS

packet from neighbor Q containing an entry to A with cost C, the router knows that it can reach A via Q with cost C.

However, the method in which each algorithm initiate path computation and propagate routing messages are very different. In Scout, path computation to node A is always initiated by A, via flooding Scout messages. However, in DV, path computation to a node can be initiated by any router. For example, in the event of a link failure, routers connected to the failed link will initiate a path computation to every node whose path used the failed link.

In addition to the differences in initiating path computations, the two algorithms also differ in how routing messages are propagated. In Scout, a Scout message contains path information only to the initiating node. Whereas in DV, a DV packet may contain path information to many nodes. The combination of path aggregation and computation initiation allows DV to efficiently compute all-pairs shortest path. As noted in Chapter 3, computing all-pairs shortest path is not as efficient using Scout.

A.2 Scout Proof of Correctness

The proof of Scout shortest-path convergence for the unrestricted Scout algorithm proceeds by showing that sink trees rooted at R are iteratively transformed into shortest-path trees rooted at R in a number of rounds bounded by the structure of the network. Intuitively this follows because every node eventually receives a Scout from the neighbor with the lowest cost to R, and therefore every node eventually knows the shortest path to R. As stated in Chapter 2, the Scout convergence and its proof apply only in situations where the network state (e.g. topology) changes at a slower rate than Scout's convergence time.

We define SPT as the shortest-path tree rooted at R. SPT has depth K, and SPT_i as the subtree of SPT, also rooted at R, such the depth of SPT_i is i. T_i is the sink tree built from R's broadcast tree built by the Scout algorithm in broadcast interval i. Tree A contains tree B iff B is a subtree of A and A and B share the same root. We prove that $\forall i \in [0, \ldots, K], T_i$ contains SPT_i.

RED HAT 1012
RED HAT V. CAS

Lemma 1: Assuming T_i contains SPT_i, then for all nodes P $\in T_i \cap SPT_i$, P will not change its forwarding tables in subsequent trees $T_j, j > i$.

Proof: Since node P $\in SPT_i$, P already has the shortest path (in cost and next-hop) to the sink R. Thus in subsequent broadcast rounds of R, P cannot receive a Scout with a lower cost, in fact, P will always receive the same lowest cost from the same neighbor (its designated neighbor). Therefore P will not alter its forwarding table (both cost and next-hop) in subsequent broadcast rounds.

This implies nodes of distance i or less from the root R in T_i will stay in the same position in T_{i+1}.

Theorem 1: Broadcast tree containment: $\forall i \in [0, \ldots, K]$, SPT_i is contained in T_i.

Proof: by induction on the depth of the broadcast tree T_i.

Base: $i = 0$. $SPT_0 = R$, and the root of T_0 is R by definition of sink trees, therefore T_0 contains SPT_0.

Induction: Assume $\forall w < z < K, T_w$ contains SPT_w, proof that T_{z+1} also contains SPT_{z+1}.

Let the set m_z = leaf nodes of(SPT_i). By the induction hypothesis and the algorithm structure, the set of nodes m_z know that in round $z + 1$, they should wait for their parent node in T_z, who will offer them the shortest cost to R. Thus in broadcast interval $z + 1$, nodes in m_z forward only R's least cost Scout to their neighbors. This implies neighbors of node $n \in m_z$ are guaranteed to receive n's shortest path to R. In particular, leaf nodes L = leaf nodes of(SPT_{z+1}) will receive the minimal path cost from nodes in m_z. After the broadcast interval $z + 1$, nodes $l \in L$ deduces the minimal path to R is through its parent node in m_z (by the definition of SPT_{z+1}). So the leaf nodes of T_{z+1} are attached in to the same nodes in SPT_{z+1}. And from lemma 1, interior nodes in T_{z+1} preserve the same connections as nodes in T_z; Therefore, T_{z+1} contains SPT_{z+1}.

From the induction when $z = K - 1$, T_K will contains SPT_K, and $SPT_K = SPT$. Since SPT contains all nodes in the network, by the tree containment definition, T_K must equal SPT. Thus after K broadcast rounds of R, the Scout algorithm is guaranteed to have

RED HAT 1012
RED HAT V. CAS

computed the shortest path to R. When $z > K$, T_z obviously contains SPT from lemma 1.

Intuitively, non-uniform link costs and link delays cause the Scout algorithm more broadcast intervals to convergence because a link can advertise a high cost in the reverse direction (to R) but have very fast forward propagation (away from R). When this occurs, node P will initially flood the costly Scout. The extra number of broadcast intervals needed to send the minimal cost Scout to P is **exactly** the shortest distance (in links) from P to an ancestor of P on the shortest-path tree who currently knows the minimal cost to R, which is always bounded by K.

Theorem 2: Shortest Path Convergence. The unrestricted Scout algorithm in Figure A.3 computes the single shortest path between every pair of nodes in the network.

Proof: This follows directly from Theorem 1. We proved that the broadcast tree eventually contains the shortest path tree for any given node sending Scouts. Thus when every node in the network generates Scouts, then in a bounded number of rounds, every node will know the shortest path to every other node.

Theorem 3: Scout message bound on convergence. The unrestricted Scout algorithm in Figure A.3 converges to the single shortest path on $O(K)$ BI's, with no more than $O(L)$ messages per round per node. Where L is the number of links in the network.

Proof: The first bound on the number of BI's for shortest path convergence is proven above. The second follows directly from the algorithm. Since every nodes can only forward one Scout per round, at most $O(L)$ Scouts can be forwarded per round per node.

A.3 Scout Summary

This chapter presented and proved the correctness of Scout, a destination initiated, selective shortest path routing algorithm. A destination initiates Scout path computation to itself by flooding Scout messages, which discovers path back to the initiating destination. Furthermore, Scout messages are small, fixed-sized and therefore can be hop-by-hop piggybacked onto data packets, largely defraying their costs to the network.

The main features of the Scout routing algorithm are

RED HAT 1012
RED HAT V. CAS

1. Independent and uncorrelated path computation

2. Destination controlled and initiated Scout messages

Scout is independent because Scout messages initiated by a destination calculate paths ONLY to that destination and is uncorrelated because the time in which a destination initiates its Scouts is determined only by the destination. Thus, Scout path computations to different destinations will occur at different times. This is in contrast to traditional LS and DV routing algorithms where all-pairs path computation occur at the same time.

Because Scout is destination initiated, the amount of Scouts injected into the network is determined by destinations. On the other hand, LS and DV's routing traffic is triggered by network changes. The advantage of this triggering property is that in a relatively static network environment, very little routing overhead is incurred. The disadvantage is that during high rates of network changes, the LS and DV routing overhead are hard to predict and control. With Scout, the amount of routing messages are not triggered and thus do not depend on network changes. This non-triggering in Scout implies that during low rates of network change, many Scout messages are sent unnecessarily (because the shortest paths did not change); On the other hand, it also means that during high rates of change, Scout messages are predictable and can be easily controlled.

The principal disadvantage of the Scout algorithm is that individual Scout messages do not aggregate path computation; thus computing all-pairs shortest paths in a relatively network static environment (such as the Internet) requires more routing message overhead than LS or DV routing algorithms. Again, the reason LS and DV performs better in this scenario is that 1) path computation is aggregated (the two algorithms always computes all-pairs shortest path) and 2) path computation are triggered by network changes (a relative static network has infrequent network changes).

RED HAT 1012
RED HAT V. CAS

Bibliography

[1] Intermediate System to Intermediate System intra-domain routing exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service, ISO 8473. ISO DP 10589, February 1990.

[2] Internet traffic archive traces. http://ita.ee.lbl.gov/html/traces.html.

[3] PNNI draft specification. ATM Forum 94-0471R13, 1994.

[4] B. Acevedo, L. Bahler, E. N. Elnozahy, V. Ratan, and M. E. Segal. Highly available directory services in DCE. In *In Proceedings of the Twenty-Sixth Annual International Symposium on Fault-Tolerant Computing (FTCS-26)*, pages 387–391, June 1996.

[5] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows.* Prentice-Hall Inc., Englewood Cliffs, NJ, 1993.

[6] V. Ahuja. Routing and flow control in systems network architecture. *IBM Systems Journal*, 18(2):298–314, 1979.

[7] G. Alkin and F. Baker. RIP version 2 MIB extension. RFC 1724, Xylogics, Inc., Cisco Systems, November 1994.

[8] Tom Anderson and John Zahorjan. Detour research homepage, 1999. http://www.cs.washington.edu/research/networking/detour/.

[9] G. Apostolopoulos, R. Guerin, and S. Kamat. Implementation and performance measurements of QoS routing extensions to OSPF. In *Proceedings of IEEE INFOCOM*, April 1999.

RED HAT 1012
RED HAT V. CAS

[10] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi. On reducing the processing cost of on-demand QoS path computation. *Journal of High Speed Networks*, 7(2):77–98, 1998.

[11] George Apostolopoulos, Roch Guerin, Sanjay Kamat, and Satish K Tripathi. Quality of Service routing: A performance perspective. In *Proceedings of ACM SIGCOMM*, pages 17–28, September 1998.

[12] Mohit Aron. Analysis of TCP Performance over ATM Networks. Master's thesis, Rice University, December 1997. http://www.cs.rice.edu/~aron/papers/ms-thesis.ps.

[13] A. A. Assad. Models for rail transportation. In *Transportation Research*, pages 205–220, 1980.

[14] A. A. Assad. Solving linear multicommodity flow problems. In *Proceedings IEEE International Conference on Circuits and Computers*, pages 157–161, 1980.

[15] S. C. Baade. SNA route generation using traffic pattern. *IBM Systems Journal*, 30(3):250–258, 1991.

[16] Saewoong Bahk and Magda El Zarki. Dynamic multi-path routing and how it compares with other dynamic routing algorithms for high speed wide area networks. In *Proceedings of ACM SIGCOMM*, pages 53–64, 1992.

[17] Sandeep Bajaj, Lee Breslau, and Scott Shenker. Uniform versus priority dropping in layered video. In *Proceedings of ACM SIGCOMM*, pages 131–143, September 1998.

[18] Hari Balakrishnan, Venkata N. Padmanabhan, and Randy H. Katz. The Effects of Asymmetry on TCP Performance. In *Proceedings of 3rd ACM Conference on Mobile Computing and Networking*, September 1997.

[19] Sara Basse. *Computer Algorithms*. Addison-Wesley Publishing Company, Reading, Mass., 1988.

RED HAT 1012
RED HAT V. CAS

[20] R. Beckers, J. L. Deneuborg, and S. Goss. Trails and U turns in the selection of a path by the ant *lasius niger*. *Journal of Theoretical Biology*, 159:397–415, 1992.

[21] Jochen Behrens and J. J. Garcia-Luna-Aceves. Distributed, scalable routing based on Link-State vectors. In *Proceedings of ACM SIGCOMM*, pages 136–147, 1994.

[22] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, N.J., 1957.

[23] Dimitri Berksekas and Robert Gallager. *Data Networks, Second Edition*. Prentice-Hall, Englewood Cliffs, N.J., 1992.

[24] D. Bertsekas and R. Gallagher. *Data Networks*. Prentice-Hall, Englewood Cliffs, CA., 1987.

[25] D. P. Bertsekas and J. N. Tsitsiklis. Parallel and distributed computation: Numerical methods. Prentice-Hall, Englewood Cliffs, N.J., 1989.

[26] S. Bradner and A. Mankin. The recommendation for the next generation IP, January 1995. Internet Request for Comments (RFC) 1752.

[27] Lawrence Brakmo and Larry Peterson. TCP Vegas: End-to-end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–1480, October 1995.

[28] Lawrence S. Brakmo, Andrew C. Bavier, and Larry L. Peterson. *x*-Sim user's manual, 1996. http://www.cs.arizona.edu/classes/cs525/xsim/xsim.html.

[29] Lee Breslau and Scott Shenker. Best-effort versus reservation: A simple comparative analysis. In *Proceedings of ACM SIGCOMM*, pages 3–16, September 1998.

[30] V. Cerf and R. Kahn. A protocol for packet network intercommunication. *IEEE Transactions on Communications*, 22(5):637–648, May 1974.

RED HAT 1012
RED HAT V. CAS

[31] Chi-Tsong Chen. *Linear System Theory and Design*. The Dryden Press, Saunders College Publishing, 1984.

[32] Johnny Chen, Peter Druschel, and Devika Subramanian. A new approach to routing with dynamic metrics. Technical report, Rice University. CS-TR 98-321, 1998.

[33] Johnny Chen, Peter Druschel, and Devika Subramanian. A simple, practical distributed multi-path routing algorithm. Technical report, Rice University. CS-TR 98-320, 1998.

[34] S. Chen and K. Nahrsted. An overview of Quality of Service Routing for next-generation high-speed networks: problems and solutions. *IEEE Networks*, pages 64–79, November/December 1998.

[35] C. Cheng. A loop-free extended Bellman-Ford routing protocol without bouncing effect. *ACM Computer Communication Review*, 19(4):224–236, 1989.

[36] C. Cheng, S. P. R. Kumar, and J. J. Garcia-Luna-Aceves. A distributed algorithm for finding K disjoint paths of minimum total length. In *In Proceedings of the 28th Annual Allerton Conference on Communication, Control, and Computing, Urbana, Illinois*, Oct. 1990.

[37] A. A. Chien and J. H. Kim. Planar-adaptive routing : low-cost adaptive networks for multiprocessors. *Journal of the Association of Computing Machinery*, 42(1):91–123, Jan 1995.

[38] B. Chinoy. Dynamics of Internet Routing Information. In *Proceedings of ACM SIGCOMM*, pages 45–52, September 1993.

[39] Isreal Cidon and Raphael Rom. Multi-path routing combined with resource reservation. In *Proceedings of IEEE INFOCOM*, pages 92–100, 1997.

RED HAT 1012
RED HAT V. CAS

[40] David D. Clark, Scott Shenker, and Lixia Zhang. Supporting real-time applications in an integrated services packet network. In *Proceedings of ACM SIGCOMM*, pages 14–26, August 1992.

[41] R. Coltun. OSPF: An Internet routing protocol. *ConneXions*, 3(8):19–25, 1989.

[42] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Mass., 1991.

[43] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. A framework for QoS-based routing in the Internet. Internet Draft, QoS Routing Working Group, Internet Engineering Task Force, expires October 1998.

[44] J. D. Day. The (un)revised OSI reference model. *Computer Communication Review*, 25:39–55, Oct. 1995.

[45] Jose Augusto de Azevedo, Joaquim Joao E. R. Silvestre, Madeira Ernesto Q. Vieira Martins, and Filipe Manuel A. Pires. A computational improvement for a shortest paths ranking algorithm. *European Journal of Operational Research*, 73:188–191, 1994.

[46] Stephen E. Deering. IP multicast and the MBone: Enabling live, multiparty, multimedia communication on the Internet. Internet-Draft, draft-ietf-manet-aodv-00.txt, November 1997. Work in progress.

[47] Stephen E. Deering. Multicast routing in internetworks and extended LANs. In *Proceedings of ACM SIGCOMM*, pages 89–101, 1988.

[48] Mikael Degermark, Andrej Brodnik, Svante Carlsson, and Stephen Pink. Small forwarding tables for fast routing lookups. In *Proceedings of ACM SIGCOMM*, pages 3–14, September 1997.

RED HAT 1012
RED HAT V. CAS

[49] A. DeHon, T. Knight, and H. Minsky. Fault-tolerant design for multistage routing networks. In *Proceedings of the International Symposium on Shared Memory Multiprocessing*, pages 60–71, 1991.

[50] Andrer DeHon, Frederic Chong, Matthew Becker, Eran Egozy, Henry Minsky, Samuel Peretz, and Tomas Knight Jr. METRO: A router architecture for high-performance, short-haul routing networks. *Proceedings the 21st Annual International Symposium on Computer Architecture*, pages 266–77, 1994.

[51] M. DeMarco and A. Pattavina. Distributed routing protocols for ATM extended banyan networks. *IEEE Journal on Selected Areas in Communications*, 15(5):925–37, June 1997.

[52] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queuing algorithm. In *Proceedings of ACM SIGCOMM*, pages 1–12, 1989.

[53] Roy C. Dixon and Daniel A. Pitt. Addressing, bridging and source routing. *IEEE Networks*, 2(1):25–32, Jan 1988.

[54] E. W. Djikstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1, 1959.

[55] J. T. Draper and J. Ghosh. Multipath E-cube algorithms (MECA) for adaptive wormhole routing and broadcasting in K-ary N-cubes. In *Sixth International Parallel Processing Symposium*, pages 470–10, 1992.

[56] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, and Satish K. Tripathi. Signal stability based adaptive routing (SSA) for ad-hoc mobile networks. *IEEE Personal Communications*, February 1997.

[57] J. B. Postel (editor). Internet Protocol. Internet Request For Comments (RFC) 729, September 1981.

RED HAT 1012
RED HAT V. CAS

[58] David Eppstein. Finding the k shortest paths. In *Proc. 35th Symp. Foundations of Computer Science*, pages 154–165. Inst. of Electrical & Electronics Engineers, November 1994.

[59] Deborah Estrin, Yakov Rekhter, and Steven Hotz. Scalable inter-domain routing architecture. In *Proceedings of ACM SIGCOMM*, pages 40–52, 1992.

[60] Michalis Faloutsos, Anindo Banerjea, and Rajesh Pankaj. QoSMIC: Quality of Service Sensitive Multicast Internet ProtoCol. In *Proceedings of ACM SIGCOMM*, pages 144–153, September 1998.

[61] Domenico Ferrari, Anindo Banerjea, and Hui Zhang. Network support of multimedia: A discussion of the tenet approach. *Computer Networks and ISDV Systems*, 10:1267–1280, July 1994.

[62] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, N.J., 1962.

[63] L. Fratta, M. Garla, and L. Kleinrock. The flow deviation method: An approach to store-and-forward communication network design. *IEEE Networks*, 3(2):97–133, 1973.

[64] L. Fratta, M. Gerla, and L. Kleinrock. Computer communication network design - experience with theory and practice, 1972. AFIPS conference proceedings SJCC, Atlantic City, New Jersey.

[65] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless interdomain routing (CIDR): An address assignment and aggregation strategy. RFC 1519, September 1993.

[66] J. J. Garcia-Luna-Aceves. A unified approach for loop-free routing using link states or distance vectors. *ACM Computer Communication Review*, 19(4):212–223, 1989.

RED HAT 1012
RED HAT V. CAS

[67] Marianne L. Gardner, Ina S. Loobeek, and Stephen N. Cohn. Type of service routing with loadsharing. In *IEEE/IECE Global Telecommunications Conference*, pages 1144–50, 1987.

[68] P. Georgatsos and D. Griffin. A management system for load balancing through adaptive routing in multiservice ATM networks. In *Proceedings of IEEE INFOCOM*, pages 863–870, 1996.

[69] R. Govindan and A. Reddy. An analysis of interdomain topology and route stability. In *Proceedings of IEEE INFOCOM*, April 1997.

[70] J. P. Gray and T. B. McNeill. SNA multiple-system networking. *IBM Systems Journal*, 18(2):263–297, 1979.

[71] R. Guerin, S. Kamat, V. Peris, and R. Rajan. Scalable QoS provision through buffer management. In *Proceedings of ACM SIGCOMM*, pages 29–40, September 1998.

[72] R. Guerin, A. Orda, and D. Williams. QoS routing mechanisms and OSPF extensions. In *Proceedings of the IEEE GLOBECOM*, November 1997.

[73] R. Guerin, D. Williams, and A. Orda. QoS routing mechanisms and QSPF extensions. In *Proceedings of the IEEE GLOBECOMM*, November 1997.

[74] C. Hedrick. Routing information protocol, 1988. Internet Request for Comments (RFC) 1058.

[75] C. Hedrick. Use of OSI IS-IS for routing in TCP/IP and dual environments, 1988. Internet Request for Comments (RFC) 1195.

[76] M. A. Henrion, G. J. Eilenberger, G. H. Petit, and P. H. Parmentier. A multipath self-routing switch. *IEEE-Communications Magazine*, 31(4):46–52, April 1993.

[77] W. C. Lee M. G. Hluchyj and P. A. Humblet. Routing subject to Quality of Service constraints in integrated communications networks. *IEEE Networks*, pages 46–55, July 1995.

RED HAT 1012
RED HAT V. CAS

[78] Janey C. Hoe. Improving the start-up behaviour of a congestion control scheme for TCP. In *Proceedings of ACM SIGCOMM*, 1996.

[79] B. Holldobler and E. O. Wilson. *Journey to the Ants*. Bellknap Press/Harvard University Press, 1994.

[80] International Standards Organization. Intra-domain IS-IS routing protocol. ISO/ICE JTC1/SC6 WG2 N323, Sept, 1989.

[81] K. Ishida, Y. Kakuda, and T. Kikuno. A routing protocol for finding two node-disjoint paths in computer networks. In *Internation Conference on Network Protocols*, pages 340–347, Nov. 1992.

[82] A. Itah and M. Rodeh. The multi-tree approach to reliability in distributed networks. In *Proceedings of the 25th Symposium on FOCS*, 1984.

[83] Van Jacobson. Congestion avoidance and control. In *Proceedings of ACM SIGCOMM*, pages 314–32, August 1988.

[84] Van Jacobson. Berkeley TCP evolution from 4.3-Tahoe to 4.3-Reno. In *Proceedings of the Eighteenth Internet Engineering Task Force*, August 1990.

[85] J. M. Jaffe, F. H. Moss, and R. A. Weingarten. SNA routing: Past, present, and possible future. *IBM Systems Journal*, 22(4):417–434, 1983.

[86] Sugih Jamin, Peter Danzig, Scott Shenker, and Lixia Zhang. A measurement-based admission control algorithm for integrated services packet networks. *IEEE/ACM Transactions on Networking*, 5(1):56–70, February 1997.

[87] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

RED HAT 1012
RED HAT V. CAS

[88] Youn Chan Jung and Chong Kwan Un. Banyan multipath self-routing ATM switches with shared buffer type switch elements. *IEEE-Transactions on Communications*, 43(11):2847–57, Nov 1995.

[89] J. L. Kennington and R. V. Helgason. *Algorithms for Network Programming*. Wiley-Interscience, New York, 1980.

[90] S. Keshav and R. Sharma. On the efficient implementation of fair queuing. *Journal of Internetworking: Research and Experience*, 2(3), September 1991.

[91] S. Keshav and R. Sharma. Achieving quality of service through network performance management. In *Proceedings of NOSSDAV*, July 1998.

[92] S. Keshav and R. Sharma. Issues and trends in router design. *IEEE-Communications Magazine*, May 1998.

[93] Srinivasan Keshav. *An Engineering Approach to Computer Networking*. Addison-Wesley Publishing Company, Reading, Mass., 1997.

[94] Atul Khanna and John Zinky. The revised ARPANET routing metric. In *Proceedings of ACM SIGCOMM*, pages 45–56, September 1989.

[95] E. Knightly and H. Zhang. D-BIND: an accurate traffic model for providing QoS guarantees to VBR traffic. *IEEE/ACM Transactions on Networking*, 5(2):219–231, April 1997.

[96] Ram Krishnan and John A. Silvester. Choice of allocation granularity in multipath source routing schemes. In *Proceedings of IEEE INFOCOM*, pages 322–329, 1993.

[97] Satish Kumar, Pavlin Radoslavov, David Thaler, Cengiz Alaettinoglu, Deborah Estrin, and Mark Handley. The MASC/BGMP architecture of Inter-domain multicast routing. In *Proceedings of ACM SIGCOMM*, pages 94–104, September 1998.

RED HAT 1012
RED HAT V. CAS

[98] Craig Labovitz, G Robert Malan, and Farnam Jahanian. Internet routing instability. Technical report, University of Mechigan. CSE Technical Report 322-97, 1997.

[99] Craig Labovitz, G. Robert Malan, and Farnam Jahanian. Internet routing instability. In *Proceedings of ACM SIGCOMM*, pages 53–64, 1997.

[100] W. S. Lai. Bifurcated routing in computer networks. *Computer Communication Review*, 15(3):28–49, 1986.

[101] T. V. Lakshman and D. Stiliadis. High speed policy-based packet forwarding using efficient multi-dimensional range matching. In *Proceedings of ACM SIGCOMM*, pages 203–214, September 1998.

[102] E. J. Lawler. *Combinatorial optimization: networks and matroids*. Holt, Rinehart and Winston, 1976.

[103] E. L. Lawler. A procedure for computing the K best solutions to discrete optimization problems and its applications to the shortest path problem. *Management Science*, 18:401–405, 1972.

[104] Eun Seol Lee and Chae Tak Lim. A study on the DRF multipath multistage interconnection network. *Journal of the Korea Institute of Telematics and Electronics*, 27(10):1605–12, Oct 1990.

[105] F. C. Lin and F. H. Wang. Message pattern routing in hypercubes: a distributed-concentrate approach. *Journal of Parallel and Distributed Computing*, 29(1):27–42, Aug 1995.

[106] Neng Pin Lu and Chung Ping Chung. A fault-tolerant multistage combining network. *Journal of Parallel and Distributed Computing*, 34(1):14–28, April 1996.

[107] Q. Ma and P. Steenkiste. On path selection for traffic with bandwidth guarantees. In *Proceedings of IEEE International Conference on Network Protocols*, October 1997.

RED HAT 1012
RED HAT V. CAS

[108] Udi Manber. *Introduction to Algorithms.* Addison-Wesley Publishing Company, Reading, Mass., 1989.

[109] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. Request for comments (RFC) 2018: TCP selective acknowledgment options, October 1996.

[110] J. M. McQuillan, I. Richer, and E. C. Rosen. The new routing algorithm for the ARPANET. *IEEE Transactions on Communications*, COM-28(5):711–719, 1980.

[111] J. M. McQuillan and D. C. Walden. The ARPANET design decisions. *Computer Networks*, 1, 1977.

[112] D. Mills. Exterior Gateway Protocol formal specification. Request for Comments (RFC) 904, April, 1984.

[113] S. P. Morgan. Queuing disciplines and passive congestion control in byte-stream networks. *IEEE Transactions on Communications*, 39(7):1097–1106, 1991.

[114] J. Moy. The OSPF specification. RFC 1131, October 1989.

[115] J. Moy. Multicast routing extensions for OSPF. *Communications of the ACM*, 37(8):61–66, August 1994.

[116] Andrew Myles, David B. Johnson, and Charles Perkins. A mobile host protocol supporting route optimization and authentication. *IEEE Journal on Selected Areas in Communications, special issue on "Mobile and Wireless Computing Networks"*, 13(5):839–849, June 1995.

[117] Paul Newman and G. Karlsson. IP switching and gigabit networks. *IEEE-Communications Magazine*, January 1997.

[118] R. Ogier and N. Shacham. A distributed algorithm for finding shortest pairs of disjoint paths. In *Proceedings of IEEE INFOCOM*, 1989.

RED HAT 1012
RED HAT V. CAS

[119] Teunis J. Ott and Neil Aggarwal. TCP over ATM: ABR or UBR? In *Proceedings of the ACM SIGMETRICS*, Seattle, WA, June 1997.

[120] Venkata N. Padmanabhan and Randy H. Katz. TCP Fast Start: A Technique For Speeding Up Web Transfers. In *Proceedings of the IEEE GLOBECOM*, November 1998.

[121] C. Parris, S. Keshav, and D. Ferrari. A framework for the study of pricing in integrated networks. ICSI Technical Report TR-92-016 and AT&T Bell Labs Technical Memorandum TM-920105-03, January 1992.

[122] Vern Paxson. Growth trends in wide area TCP connections. *IEEE Networks*, 8(4):8–17, 1994.

[123] Vern Paxson. End-to-end Internet packet dynamics. In *Proceedings of ACM SIGCOMM*, pages 139–152, September 1997.

[124] Vern Paxson. Measurements and analysis of end-to-end Internet dynamics, April 1997. Ph.D. dissertation, University of California, Berkeley.

[125] Vern Paxson. End-to-End routing behavior in the Internet. *IEEE/ACM Transactions on Networking*, 5(5):43–51, 1998.

[126] Vern Paxson and Sally Floyd. Wide-area traffic: the failure of poisson modeling. In *Proceedings of ACM SIGCOMM*, pages 257–268, August 1994.

[127] Charles Perkins. Ad Hoc On Demand Distance Vector (AODV) routing. Internet-Draft, draft-ietf-manet-aodv-00.txt, November 1997. Work in progress.

[128] Michael Perloff and Kurt Reiss. Improvements to TCP Performance in High-Speed ATM Networks. *Communications of the ACM*, 38(2):90–100, February 1995.

[129] Larry L. Peterson and Bruce S. Davie. *Computer Networks: a Systems Approach.* Morgan Kaufmann Publishers, Inc., San Francisco, CA., 1996.

RED HAT 1012
RED HAT V. CAS

[130] M. De Prycker. *Asynchronous Transfer Mode: solution for broadband ISDN.* Ellis Horwood, Chichester, England, 1991.

[131] N. S. V. Rao and S. G. Batsell. QoS routing via multiple paths using bandwidth reservation. In *Proceedings of IEEE INFOCOM*, pages 11–18, 1998.

[132] S. M. Reddy and V. P. Kumar. On multipath multistage interconnection networks. In *Proceedings of the 5th International Conference on Distributed Computing Systems*, pages 210–17, 1985.

[133] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771, T.J. Watson Research Center, IBM Corp., Cisco Systems, March 1995.

[134] E. C. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. *draft-itef-mpls-arch-04.txt* Internet Draft, February 1999, work in progress.

[135] H. Saran, S. Keshav, and C. R. Kalmanek. A scheduling discipline and admission control policy for Xunet 2. In *Proceedings of NOSSDAV*, November 1993.

[136] Stefan Savage, Tom Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoff Voelker, and John Zahorjan. Detour: a case for informed Internet routing and transport. Technical report, University of Washington. CS Technical Report UW-CSE-98-10-05, 1998.

[137] M. Schwartz. *Telecommunication Networks, Protocols, Modeling and Analysis.* Addison-Wesley, Reading, Mass., 1987.

[138] Mischa Schwartz and Thomas E. Stern. Routing techniques used in computer communication networks. *IEEE Transactions on Communications*, 28(4):539–552, 1980.

[139] A. Segall and J. M. Jaffe. A reliable distributed route set-up procedure. In *IEEE/IECE Global Telecommunications Conference*, 1983.

RED HAT 1012
RED HAT V. CAS

[140] Seung Woo Seo and Tse Yun Feng. The composite banyan network. *IEEE-Transactions on parallel and distributed systems*, 6(10):1043–1054, Oct 1995.

[141] D. R. Shier. On algorithms for finding the k sortest paths in a network. *Networks*, 9:195–214, 1979.

[142] R. Siamwalla, R. Sharma, and S. Keshav. Discovering Internet topology. Technical report, Cornell University. submitted to Infocom, 1999.

[143] F. Sibai and S. Kulkarni. Performance of multicast wormhole routing algorithms in fault-tolerant 2D meshes. *Seventh International Conference on Parallel and Distributed Computing Systems*, pages 610–613, 1994.

[144] F. N. Sibai and A. A. Abonamah. Parallel path assignment of multicast connections in multi-path networks. In *IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, 1994.

[145] F. N. Sibai, N. K. Sharma, and A. A. Abonamah. A simulation study of four reconfiguration algorithms for a multi-path cube-based network. *Transactions of the Society for Computer Simulation*, 10(1):1–21, March 1993.

[146] Deepinder Sidhu, Raj Nair, and Shukri Abdallah. Finding disjoint paths in networks. In *Proceedings of ACM SIGCOMM*, pages 43–51, 1991.

[147] Josep Soel-Pareta, Debapriya Sarkar, Jorg Liebeherr, and Ian F Akyildiz. Adaptive multipath routing of connectionless traffic in an ATM network. In *Proceedings of IEEE INFOCOM*, pages 1626–1630, 1995.

[148] J. Sole-Pareta, D. Sarkar, J. Liebeherr, and I. F. Akyildiz. Adaptive multipath routing of connectionless traffic in an ATM network. *Journal of Network and Systems Management*, 3(4):355–370, 1995.

RED HAT 1012
RED HAT V. CAS

[149] V. Srinivasan, George Varghese, Subash Suri, and Marcel Valdvogel. Fast scalable level four switching. In *Proceedings of ACM SIGCOMM*, pages 191–202, September 1998.

[150] Martha Steenstrup. *Routing in communications networks.* Prentice-Hall, Fort Collins, CO., 1995.

[151] Ion Stoica, Scott Shenker, and Hui Zhang. Core-stateless fair queuing: A scalable architecture to approximate fair bandwidth allocations in high speed networks. In *Proceedings of ACM SIGCOMM*, pages 118–130, September 1998.

[152] J. W. Surballe and R. E. Tarjan. A quick method of finding shortest pairs of disjoint paths. *Networks*, 4:43–51, 1984.

[153] Hiroshi Suzuki and Fouad A. Tobagi. Fast bandwidth reservation scheme with multi-link & mult-ipath routing in ATM networks. In *Proceedings of IEEE INFOCOM*, pages 2233–2240, 1992.

[154] Andrew S. Tanenbaum. *Computer Networks.* Prentice-Hall, Englewood Cliffs, NJ., 1981.

[155] S. Tiarawut, T. Saito, and H. Aida. A connection-level design of multistage nonblocking ATM switches. *IEICE-Transactions on Communications*, E77-B(10):1203–8, Oct 1994.

[156] D. M. Topkis. A k shortest path algorithm for adaptive routing in communications networks. *IEEE Transactions on Communications*, 36, 1988.

[157] J. N. Tsitsiklis and G. D. Stamoulis. On the average communication complexity of asynchronous distributed algorithms. MIT LIDS-report LIDS-P-2238, May 1986.

[158] K. Varadhan, R. Govindan, and D. Estrin. Persistent routing oscillations in Inter-domain routing. USC/ISI, Available at Routing Arbiter project's home page.

RED HAT 1012
RED HAT V. CAS

[159] C. Villamizer, R. Chandra, and R. Govindan. *draft-IETF-idr-route-dampen-00-preview*, 1996. Internet Engineering Task Force Draft, July 21, 1995.

[160] Marcel Waldvogel, George Varghese, Jon Turner, and Bernhard Plattner. Scalable high speed IP routing lookups. In *Proceedings of ACM SIGCOMM*, pages 25–36, September 1997.

[161] Mu Cheng Wang, H. J. Siegel, M. A. Nichols, and S. Abraham. Using a multipath network for reducing the effects of hot spots. *IEEE-Transactions on Parallel and Distributed Systems*, 6(3):252–68, March 1995.

[162] Z. Wang and J. Crowcroft. Quality of Service routing for supporting multimedia applications. *IEEE Jounal Selected Areas in Communications*, 14(7):1228–1234, 1996.

[163] Zheng Wang and Jon Crowcroft. Analysis of shortest-path routing algorithms in a dynamic network environment. In *ACM SIGCOMM Computer Communication Review*, pages 63–71, 1992.

[164] I. Widjaja and Leon Garcia. The helical switch: a multipath ATM switch which perserves cell sequence. *IEEE-Transactions on Communications*, 42(8):2618–29, Aug 1994.

[165] J. Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17:712–716, 1971.

[166] Zhou Yigquan and Min Yinghua. A kind of multistage interconnection networks with multiple paths. *Journal of Computer Science and Technology*, 11(4):395–404, July 1996.

[167] Lixia Zhang, Steve Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala. RSVP: A new resource reservation protocl. *IEEE Network Magazine*, 7(5):8–18, September 1993.

RED HAT 1012
RED HAT V. CAS

[168] Lixia Zhang, Scott Shenker, and David D. Clark. Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic. In *Proceedings of ACM SIGCOMM*, pages 133–148, 1991.

[169] Z. Zheng and J. Crowcroft. QoS Routing for supporting resource reservation. In *IEEE JSAC*, September 1996.

RED HAT 1012
RED HAT V. CAS

A Transport Layer Approach for Achieving Aggregate Bandwidths on Multi-homed Mobile Hosts *

Hung-Yun Hsieh and Raghupathy Sivakumar
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
{hyhsieh, siva}@ece.gatech.edu

ABSTRACT

Due to the availability of a wide variety of wireless access technologies, a mobile host can potentially have subscriptions and access to more than one wireless network at a given time. In this paper, we consider such a multi-homed mobile host, and address the problem of achieving bandwidth aggregation by striping data across the multiple interfaces of the mobile host. We show that both link layer striping approaches and application layer techniques that stripe data across multiple TCP sockets do not achieve the optimal bandwidth aggregation due to a variety of factors specific to wireless networks. We propose an end-to-end transport layer approach called *pTCP* that effectively performs bandwidth aggregation on multi-homed mobile hosts. We show through simulations that pTCP achieves the desired goals under a variety of network conditions.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms, Design, Performance

Keywords

Multi-homed mobile host, Bandwidth aggregation, Striping

1. INTRODUCTION

The explosive growth in the number of mobile Internet users has been accompanied by the equally staggering increase in the number of wireless access technologies. A mobile user today can choose from a myriad of options ranging from networks such as Globalstar or Iridium for satellite access, CDPD, GPRS, EDGE, or 3G for wide area access, Ricochet for metropolitan area access, and IEEE 802.11 or HiperLAN based networks for local area access. An interesting and obvious challenge that arises when such independent options exist is: *How can these technologies co-exist in providing the best wireless access possible to mobile users?*

As a first step towards addressing this challenge, approaches have been proposed for performing *vertical handoffs* from one network to another as the mobile user migrates across coverage areas [19]. When the coverage areas of two networks overlap (say wide area and local area), the user is provided access through the higher data rate connection. In this paper, we consider an identical scenario of a mobile host having multiple wireless interfaces. However, instead of the mobile host being provided access only through one of its interfaces, we consider the problem of providing simultaneous access through all of its active interfaces. For example, if a user is subscribed to both a wide-area wireless network with a data rate of 2Mbps indoors (e.g. 3G systems), and a local-area wireless network with an effective data rate of 5Mbps (e.g. IEEE 802.11), assuming that the user is within range of the access points of both networks we address the following question: *Can an application that requires reliable and sequenced delivery of data be provided a data rate of 7Mbps through the use of both interfaces?* Since TCP is by far the most dominant protocol used for reliable and sequenced data delivery, we use TCP in all of our discussions henceforth and use the generic term *sockets* to refer to TCP sockets.

A simple approach to aggregate bandwidths would be to use multiple sockets, one each for every active interface, and use application layer striping and resequencing. In fact, similar schemes have been proposed to improve application layer throughput, albeit in a different context where such striping is done on sockets that share a long-fat path and the goal is to fill the bandwidth-delay product of the path (which otherwise will be impossible without both ends supporting the window scaling option) [2, 17]. However, in the context of multi-homed mobile hosts, it turns out that such an approach not only fails to achieve the aggregate data rate, but in the specific case of the connections having vastly differing bandwidth-delay products can result in the effective aggregate data rate being lower than the data rate of the slowest connection! While we identify the reasons involved and discuss them in detail in Section 2, briefly the performance degradation occurs due to head-of-line blocking at the resequencing buffer of the receiving application. Pending packet arrivals at the receiving end of the slower connection stall the TCP sender of the faster connection, which eventually enters persist mode because of zero window advertisement by its receiving end [22].

Link layer striping schemes assuming stable link characteristics have been proposed earlier [6, 21] for bandwidth aggregation. However, such schemes are unfortunately inapplicable to wireless

*This work was sponsored by the National Science Foundation under award #0117840, and Yamacraw.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MOBICOM'02, September 23–26, 2002, Atlanta, Georgia, USA.
Copyright 2002 ACM 1-58113-486-X/02/0009 ...$5.00.

RED HAT 1012
RED HAT V. CAS

links. Even with an adaptive mechanism that monitors the quality of wireless links and stripes accordingly [18], optimal bandwidth aggregation still cannot be achieved in the context of multi-homed mobile hosts where different interfaces potentially belong to different access networks. In [1], the authors propose a "channel" striping algorithm where a channel is defined as a logical FIFO path at any layer of the protocol stack including the transport layer. We discuss in Section 7 why this approach exhibits drawbacks and limitations that specifically pertain to link layer striping schemes.

In this paper, our goal is to study the problems involved in achieving bandwidth aggregation when an application on a mobile host uses multiple interfaces simultaneously, and propose a transport layer approach that effectively addresses the problems. To this end, we propose a purely end-to-end transport layer approach called *pTCP* (parallel TCP) and present it as a wrapper around a slightly modified version of TCP that we refer to as *TCP-v* (TCP-virtual). For each pTCP socket opened by an application, pTCP opens and maintains one TCP-v connection for every interface over which the connection is to be striped on. pTCP manages the send buffer across all the TCP-v connections and decouples loss recovery from congestion control, performs intelligent striping of data across the TCP-v connections, does data reallocation to handle variances in the bandwidth-delay product of the individual connections, redundantly stripes data during catastrophic periods (such as blackouts or resets), and has a well defined interface with TCP-v that allows different congestion control schemes to be used by the different TCP-v connections. We show through *ns2* [15] based simulations that pTCP outperforms both simple and sophisticated schemes employed at the application layer. Although we present pTCP as a transport layer solution for simplicity, it can be implemented as a true session layer solution provided sufficient support is provided by the transport layer. The contributions of this work can thus be summarized as follows:

1. We consider mobile hosts that have multiple interfaces corresponding to independent wireless access networks, and investigate why using multiple sockets with application layer support does not result in the desired bandwidth aggregation.

2. We propose an end-to-end transport layer approach called pTCP that effectively provides applications with the aggregate bandwidths available through the multiple interfaces at a mobile host.

The rest of the paper is organized as follows: Section 2 discusses why using multiple sockets does not result in aggregate bandwidths. Section 3 presents the assumptions and key tenets of the pTCP design. Section 4 describes the pTCP approach in detail along with the pTCP state diagram, handshakes, and packet header formats. Section 5 provides simulation results of the pTCP approach. Section 6 revisits several assumptions and considers the impact of relaxing them. It also discusses several deployment issues for pTCP. Finally, Section 7 discusses related work and Section 8 concludes the paper.

2. MOTIVATION

In this section, we consider an approach wherein the application opens multiple TCP sockets, one each for every interface, and performs striping of data across the different sockets to achieve bandwidth aggregation. We consider both an *unaware* application that has no knowledge of the underlying connection data rates, and a *smart* application that has some knowledge of the available data rates (which will consequently enable it to stripe more intelligently). In the former case, when a socket blocks on a write, the sending application moves to the next socket and so on. In the latter case, the sending application stripes data based on a ratio determined by estimation of the available rates on the different connections (or pipes[1]). Since the goal is to perform reliable in-sequence data delivery, the receiving application does the resequencing using a finite resequencing buffer. For simplicity, we assume application level sequence numbers to facilitate the resequencing process, and restrict our discussions to packet streams as opposed to byte streams. The receiving application continues to read packets from each socket as long as its resequencing buffer has available space. When the application buffer is full, it stops reading from sockets that have already delivered packets with sequence numbers larger than the next expected application level sequence number. It then enters a *peek mode*[2] where it peeks into the next available packet in each of the other sockets and reads a packet only when it is the next in-sequence packet. Note that if the application buffer size is zero, the application will always read in-sequence packets from the sockets. On the other hand, increasing the size of the application buffer has the effect of reducing the chances of the faster pipe being stalled by the slower one. We elaborate on this phenomenon later in this section when we discuss the impact of data rate differential among the multiple pipes.

We now proceed to identify the key constraints of such an application layer striping approach:

- *Data Rate Differential:*

  When the data rates of the pipes used by the unaware application are different, the aggregate bandwidth achieved by the simple approach remains a tight function of the data rate of the slowest pipe. This can intuitively be explained as follows: Consider two pipes with data rates of 10Mbps and 2Mbps respectively. Since the application stripes data by keeping the send buffer of each pipe filled, a send-buffer's worth of application data will be injected to the first (10Mbps) pipe (let this block of data be B_1). Blocked by the first pipe, the application will then proceed to inject data into the second (2Mbps) pipe (let this block of data be B_2). Because the first pipe will drain data faster, the application will, after filling the second pipe, inject more data into the first pipe (let this block of data be B_3). Assume that because of the data rate difference, the first pipe delivers B_3 before B_2 is drained out by the second pipe.

  Since the additional data (block B_3) will be out-of-order, it will be queued up in the resequencing buffer of the receiving application pending the arrival of the entire block of data B_2 through the second pipe. Because the first pipe will continue to transfer data at a faster rate, this will eventually result in the application's resequencing buffer overflowing. The receiving application will thereupon stop reading data from the first pipe, which in turn will cause the first pipe's TCP receiver buffer to fill up. The TCP receiver will then advertise a window size of zero, completely stalling the first pipe. Once the in-sequence data (block B_2), sent originally through the second pipe, reaches the receiver and hence releases space in the resequencing buffer, the first pipe will become active again. Note that such head-of-line blocking is indeed an artifact of the unaware striping mechanism used

[1] We refer to the individual connections as pipes from hereon to differentiate them from the aggregate end-to-end connection.

[2] The $recv()$ socket call and its variants support a peek flag that when set allows the receive operation to retrieve data from the beginning of the receive buffer without removing that data from the buffer [22].

RED HAT 1012
RED HAT V. CAS

by the application. One way of reducing the above coupling between the faster and slower pipes is to increase the resequencing buffer size at the application layer. The larger the buffer size, the more the time for which the faster pipe can remain active without being inhibited through flow control. Specifically, if the two pipes have bandwidths of R_1 and R_2 ($R_1 < R_2$) respectively and equal delays, the application buffer required in steady state to effectively aggregate bandwidths is $\frac{R_2}{R_1} * W$, where W is the default socket buffer size. Even assuming that the above buffer requirements can be accommodated, such buffering still cannot handle stalls that occur due to losses in the slower pipe. Also, even if the application does smart striping, such a problem will exist as long as the striping ratio does not exactly match the data rate ratio of the different pipes. We elaborate on this issue in the next constraint. Furthermore, in TCP the performance degradation for the simple approach is severe because of another phenomenon: persist timers. When the sender of the faster pipe receives a window advertisement of zero, it enters persist mode. If the single *window update* from the receiver happens to be lost (either due to congestion or random wireless losses), the sender probes the receiver only after the persist timer expires next (5 seconds). The persist timer value doubles after every unsuccessful probe and is capped only at 60 seconds [22]. While this effectively brings down the progress of the faster pipe to a crawl, the impact is more severe as the slower pipe can potentially enter persist mode because of the persist-timer induced stalling of the faster pipe! Hence, in TCP the effect of the data rate differential among the different pipes can potentially be catastrophic to the application, resulting in the aggregate throughput being lower than the data rate of the slowest pipe. We present results in Section 5 that illustrate this phenomenon.

- *Fluctuating Data Rates:*

  Although the problem due to data rate differential can be overcome by employing an intelligent striping scheme, performing such intelligent striping is inherently a difficult problem because of two reasons: (i) The pipes are end-to-end pipes that traverse multiple hops between the sender and the receiver, and the available bandwidth is likely to fluctuate dynamically; and (ii) Given the dynamic nature of wireless link characteristics, it is very likely that the pipes will exhibit highly varying data rates. When the application stripes based on the estimated data rates of the pipes, and the data rates change, the very purpose of intelligent striping is defeated resulting in degraded performance. Note that the dynamic characteristics of the wireless link, and the consequent difficulty in performing accurate rate estimation is only part of the reason for the degraded performance. The coupling of congestion control and loss recovery (for the aggregate connection) that exists because of the individual TCP pipes functioning independent of each other is also a contributing factor. For example, packets assigned to a TCP pipe by an application cannot be "withdrawn" from that pipe, notwithstanding any bandwidth reduction the pipe may experience. Thus, if bandwidth reduction were to occur, packets assigned to the pipe that have not yet been transmitted due to lack of space in the reduced congestion window cannot be reassigned to another active pipe.

- *Blackouts:*

  Blackouts are extreme cases of rate fluctuations where the available data rate falls to zero and remains at zero for an extended period of time. Causes for such phenomena include temporary loss in connectivity (e.g. when the user is passing through a tunnel), fading, interference from a moving source, etc. Observations on the occurrence of such phenomena have been made in related work [16]. In the multiple sockets approach, such blackouts on one or a subset of the pipes will stall the entire aggregate connection because of buffer overflow at the receiving application. This is obviously an undesirable phenomenon. While the only solution to this problem is to have some feedback mechanism at the application layer (for the application to realize that a particular pipe has stalled), this will substantially increase the overhead and complexity in the application.

- *Application Complexity:*

  Although the above application layer approaches are simple in the sense that they do not require any protocol changes at the transport layer, the complexity and overhead at the application layer is considerable. Essentially the application has to implement a resequencing mechanism over the reordering already performed within each pipe by TCP. Sequence numbers that facilitate the resequencing have to be included in application defined headers, and the application has to explicitly ensure that the application layer "segments" (that have unique application layer sequence numbers) do not get fragmented. One conceivable way the application can ensure that application layer segments are not fragmented is to write exactly one MSS worth of data during every write. If nagling is enabled [12], this would achieve the desired goal. Similarly, in order to stripe intelligently, the application will have to redundantly implement a bandwidth estimation mechanism in spite of the bandwidth estimation already performed by TCP through its congestion control mechanism. Furthermore, in order to solve the problems identified as consequences of blackouts, the application will have to implement a feedback mechanism to recover from pipes that are stalled, and in effect duplicate both the reordering and loss recovery mechanisms already implemented by TCP for the individual pipes. It is clearly undesirable to overload applications in such a manner when all applications on the mobile host would require similar functionality. Note that the above arguments would also hold for session layer approaches in the absence of appropriate interfaces between the session layer and the transport layer.

- *Multiple Congestion Control Schemes:*

  Since different wireless network technologies possess very diverse characteristics in terms of throughput, delay, jitter, loss rates, etc., approaches to improve TCP performance over wireless networks have typically been proposed for specific scenarios. For example, while *snoop* [5] has been proposed primarily for WLANs, [16] shows that snoop is inappropriate in WWANs due its key assumption that wireless link delays are insignificant when compared to the end-to-end delays. WTCP, proposed in [16] for WWANs, however will stand inappropriate in WLANs due its reliance on inter-packet separation as the key congestion metric. WWANs have low data rates that result in the inter-packet delay being large, which in turn makes it a robust and realistic metric to use. However, in WLANs where bandwidths can be in the order of tens of megabits per second, it no longer serves as a reliable congestion metric. Similarly, approaches such as [8] to specif-

RED HAT 1012
RED HAT V. CAS

ically improve TCP's performance over satellite links that possess very large bandwidth-delay products have also been proposed. When a mobile host has multiple interfaces, a conceivable scenario (until a unified transport layer framework is derived) is one where interface-specific transport protocols will be used. In the simple application layer striping approach, besides the numerous roles assigned to the application, the task of choosing an appropriate transport protocol for the different pipes will also rely on the application.

3. THE PTCP DESIGN

In this section, we present the key design elements of the pTCP approach that overcome the drawbacks identified earlier for the application layer approaches. The following assumptions are made for the basic design of pTCP: (i) Mobile hosts have multiple interfaces, which they would ideally like to use simultaneously for a single application connection; (ii) Both the sender and the receiver support pTCP; (iii) The bandwidth bottlenecks are purely in the wireless links for the individual pipes; and (iv) The application should ideally be unaware of the striping process. Briefly, assumption (iii) is to ensure TCP-friendliness of the aggregate connection in the backbone Internet where paths of multiple pipes may merge. While we make the assumptions primarily for simplifying the presentation of pTCP, we revisit assumptions (ii) and (iii) in Section 6 and discuss the required modifications to pTCP if the assumptions are to be relaxed.

The pTCP approach is based on the following five key design elements:

- *Decoupled Congestion Control and Reliability:*

  As described in Section 1, pTCP is a *wrapper* around a modified TCP that we refer to as *TCP-v*. While we present the details of the interaction between pTCP and TCP-v in Section 4, briefly pTCP maintains and controls a single send buffer across all the TCP-v pipes for the aggregate connection. The individual TCP-v pipes perform congestion control and loss recovery just like regular TCP. However, any segment transmission by a TCP-v is preceded by an explicit call to pTCP requesting for application data. Since pTCP has control over the buffer, a retransmission at the TCP-v level does not need to be a retransmission at the pTCP level. However, the amount of data that can be sent out through each TCP-v pipe is strictly determined by the TCP congestion control algorithm employed by each respective pipe. Therefore, TCP-v controls the *amount* of data that can be sent while pTCP controls *which* data to send. In this fashion, pTCP decouples congestion control and reliability. We describe as we go along how the decoupling contributes to improved performance and functionality in pTCP.

- *Congestion Window Based Data Striping:*

  When a TCP-v pipe has space in its congestion window for transmissions, it requests pTCP for data. If there exists no unsent data, pTCP registers the concerned TCP-v pipe as an *active pipe* and returns `NO_DATA`. The TCP-v pipe then waits for a subsequent *resume* call from pTCP before requesting for data again. When pTCP receives new data from the application, it issues the resume call only to those TCP-v pipes that are registered as active. Note that such striping is different from striping that is conditional on buffer availability (as seen in the simple application layer approach). In pTCP, *data will be given to a TCP-v pipe only when there is space in its congestion window for the data to be sent.* Note that this inherently assumes the congestion window to be a true representative of the bandwidth-delay product of the pipe. While the TCP congestion window *is* an approximation of the bandwidth-delay product, it is possible that it is an incorrect estimation (say, for example, due to deep buffers in the network). We revisit this issue in Section 6. The striping of data based on the congestion window of the individual pipes removes the problem that arises due to differences in the rates of the pipes, *provided there is no fluctuation in the available bandwidth.*

- *Dynamic Reassignment during Congestion:*

  Recall that it is possible for the congestion window to be an over-estimate especially just before congestion occurs. This can result in an undesirable hold up of data in pipes where the congestion window was reduced recently. For example, consider a scenario in which the congestion window of pipe p_i is $cwnd_i$. If $cwnd_i$ worth of data is assigned to p_i, and the window is cut down to $\frac{cwnd_i}{2}$ due to bandwidth fluctuations, the $\frac{cwnd_i}{2}$ worth of data that falls outside the congestion window of p_i will be blocked from transmission till the $cwnd_i$ opens up. In the meantime, this is equivalent to a static scenario in which the application undesirably assigned more data than what a pipe can carry and in the process slows down other faster pipes. pTCP solves this problem by leveraging the decoupling that exists between congestion control and reliability. When a pipe experiences congestion, irrespective of whether the detection is through duplicate acknowledgements or a timeout, the window is reduced (by half in the former and to one in the latter). If the congestion window of a pipe is thus reduced, pTCP immediately *unbinds* the data that was bound to the sequence numbers of the concerned pipe that fall outside the current congestion window. Thus, if another pipe has space in its congestion window and requests for data, the unbound data is now available for reassignment to that pipe. When the original pipe requests for data corresponding to the same sequence number that was unbound, new application data is bound by pTCP and returned to the pipe. Such a reassignment strategy greatly improves the performance of pTCP under dynamic conditions, as we illustrate through simulation results in Section 5. The trade-offs of the reassignment strategy is the potential overhead of performing unnecessary retransmissions. Our simulation results show that this overhead is insignificant.

- *Redundant Striping for Blackouts:*

  While the strategy described above reassigns data that falls out of a pipe's congestion window, it does not deal with the one MSS worth of data (the first MSS in the congestion window) that will never fall out of the congestion window irrespective of the state of the pipe. Failure to deliver that one MSS worth of data can potentially stall the entire aggregate connection if the concerned pipe undergoes multiple timeouts or suffers a blackout. Hence, pTCP *redundantly stripes the first MSS of data in a congestion window that has suffered a timeout, onto another pipe.* In doing so, the binding of the data is changed to the new pipe, although the old pipe has access to a copy of the same data. The reason for leaving a copy behind instead of a regular reassignment is that the old pipe will require at least one MSS worth of data to send in order to recover. At the same time, providing it with a new MSS worth of data is a potential pitfall because of the

RED HAT 1012
RED HAT V. CAS

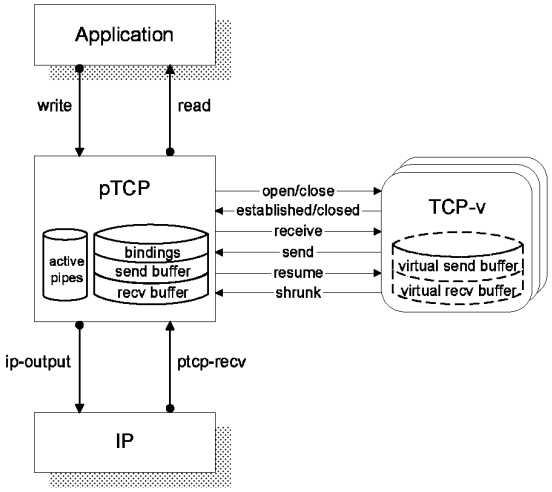


Figure 1: Overview of pTCP and Key Data Structures

chances of blocking, given that the pipe is experiencing severe conditions.

- *Selective Acknowledgments:*

  The pTCP design does not impose any requirements on the design of the TCP-v protocol used by the individual pipes. However, the use of TCP-SACK helps the performance of pTCP under certain conditions. We present here an argument for why TCP-v in pTCP should preferentially use TCP-SACK. When there are multiple losses within a congestion window, TCP-Reno and TCP-NewReno will recover from the losses at the rate of one loss per round-trip time. Thus, if multiple losses occur within a congestion window of a pipe, the time taken to recover from a loss farther down in the congestion window increases. In default TCP, this is acceptable as the alternative is to take a pessimistic attitude like in TCP-Tahoe that by default treats all packets after a hole to be lost, and hence starts retransmitting them. However, in pTCP, such a delayed recovery can make the hole potentially stall the entire aggregate connection. This makes the rate at which loss recovery is done critical. When TCP-SACK is used, loss recovery is done faster with multiple holes filled within one round-trip time because of the SACK information exchanged. This results in pTCP experiencing better performance. Since TCP-SACK is preferred in a general Internet setting [11], and more so in wireless environments to enable faster recovery from random channel errors [16], we believe that the recommendation for the use of SACK in pTCP is a reasonable one.

4. THE PTCP PROTOCOL

4.1 Overview

Figure 1 provides an architectural overview of the pTCP approach. pTCP acts as the central engine that interacts with the application, IP, and TCP-v respectively. For each interface used by the application to achieve bandwidth aggregation, pTCP creates and maintains one TCP-v pipe. *We assume that the choice of the number of interfaces to use is an external decision, and is conveyed to pTCP through a socket option.* The figure also illustrates the key data structures maintained for every aggregate connection. pTCP controls and maintains the send and receive socket buffers for the connection. Application data writes are served by pTCP, and the data is copied onto the `send_buffer`. A list of active TCP-v pipes (that have space in the congestion window to transmit) called `active_pipes` is maintained by pTCP. A TCP-v pipe is placed in `active_pipes` initially when it is created by pTCP. Upon the availability of data that needs to be transmitted, pTCP sends a $resume()$ command to the active TCP-v pipes. Once a resume is issued to a pipe, the corresponding pipe is removed from `active_pipes`. A TCP-v pipe that receives the command builds a regular TCP header based on its state variables and gives the segment (sans the data) to pTCP through the $send()$ interface. pTCP binds an unbound data segment in the `send_buffer` to the header of the "virtual" segment TCP-v has built, maintains the binding in the data structure called `bindings`, appends its own header and sends it to the IP layer. A *resumed* TCP-v continues to issue $send()$ calls till there is no more space left in the congestion window, or pTCP responds back with a `NO_DATA` return value to *freeze* the concerned pipe (note that the TCP-v pipe needs to perform a few rollback operations to account for the unsuccessful transmission). When pTCP receives a $send()$ call, and has no unbound data left, it returns a `NO_DATA` value, and adds the corresponding pipe to `active_pipes`.

When pTCP receives an ACK, it strips the pTCP header, and hands over the packet to the appropriate TCP-v pipe (through the $receive()$ interface). The correct TCP-v pipe is recognizable from the TCP 4-tuple. The TCP-v pipe processes the ACK in the regular fashion, and updates its state variables including the virtual send buffer. The virtual buffer can be thought of as a list of segments that have only appropriate header information. The virtual send and receive buffers are required to ensure regular TCP semantics for congestion control and connection management within each TCP-v pipe. The pTCP header carries cumulative pTCP level ACK information that pTCP uses to purge its receive buffer if required. When an incoming data segment is received by pTCP, it strips both the pTCP header and the data, enqueues the data in the `recv_buffer`, and provides the appropriate TCP-v with only the skeleton segment that does not contain any data. TCP-v treats the segment as a regular segment except that no application data is queued in the virtual receive buffer.

In the rest of the section, we describe the different components of the pTCP protocol including interfaces with TCP-v, header formats, connection management, and congestion and flow control.

4.2 TCP-v Interface

As seen in Figure 1 the following eight functions act as the interface between pTCP and TCP-v: $open()$, $close()$, $established()$, $closed()$, $receive()$, $send()$, $resume()$, and $shrunk()$. pTCP uses the $open()$ and $close()$ calls as inputs to the TCP-v state machine for opening and closing a TCP-v pipe respectively. TCP-v uses the $established()$ and $closed()$ interfaces to inform pTCP when its state machine reaches the **ESTABLISHED** and **CLOSED** states respectively [22]. The $send()$ call is used by TCP-v to send "virtual" segments to pTCP which will then bind the segments to real data. The $receive()$ interface on the other hand is used by pTCP to deliver "virtual" segments to TCP-v. pTCP uses $resume()$ to inform TCP-v that additional unbound data is available. TCP-v, upon receiving the call, attempts to send as much data as possible till it gets a `NO_DATA` return value on its $send()$ call and freezes. Finally, TCP-v uses the $shrunk()$ interface to inform pTCP of any change in its congestion window so that pTCP can perform reassignment as described in Section 3.

RED HAT 1012
RED HAT V. CAS

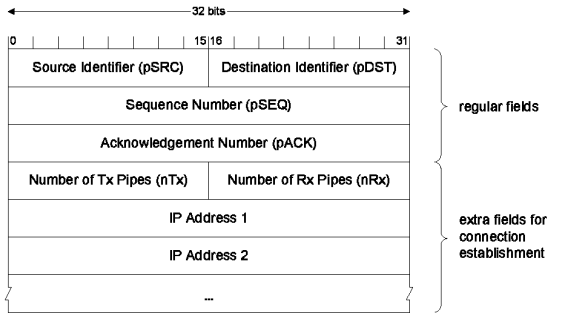


Figure 2: pTCP Header Format

4.3 Header Formats

Figure 2 presents the header formats for the pTCP protocol. Note that the header is in addition to the regular TCP header that will be used by TCP-v. The regular pTCP header consists of the following four fields: (i) source connection identifier ($pSRC$), (ii) destination connection identifier ($pDST$), (iii) pTCP sequence number ($pSEQ$), and (iv) pTCP acknowledgement number ($pACK$). The connection identifiers are used to uniquely identify the aggregate pTCP connection at both ends. The $pSEQ$ is the sequence number at the aggregate connection level and is independent of the TCP sequence number. The $pACK$ is a cumulative acknowledgement similar to the TCP acknowledgement field. Since the individual TCP-v pipes will use the TCP ACK fields to perform congestion control (recall that congestion control and loss recovery are coupled in TCP), they cannot be reused by pTCP. Because pTCP is responsible for performing flow control (given that it controls the buffer), it requires a field for window advertisement as in TCP. However, since TCP-v pipes do not have to perform flow control (they merely maintain virtual buffers), pTCP reuses and overrides the TCP window advertisement field for performing flow control. The reuse does not interfere with the progress of the individual TCP-v pipes due to the fact that the pTCP advertised window will always be greater than the actual window of an individual pipe (we elaborate on this in Section 4.5).

In addition to the regular pTCP header fields, the header format for the connection establishment phase is further augmented with the following fields: (i) number of transmitting interfaces to be desirably used (nTx), (ii) number of receiving interfaces that can be used (nRx), (iii) list of IP addresses corresponding to nTx ($ipTx$), and (iv) list of IP addresses corresponding to nRx ($ipRx$). The nTx field is the number of interfaces the source would ideally like to use for its transmissions (which in effect will require nTx pipes to be maintained at both ends), and the nRx field is the maximum number of interfaces on which the source is willing to serve the reverse path. Note that even if nRx is 1 at one end, the other end can still use multiple interfaces, but all pipes will terminate in the one interface at this end. (This would be the typical setup when a multi-homed mobile host communicates with an Internet backbone host that has only one interface.)

4.4 Connection Management

We use the state machine of pTCP and the connection establishment handshake presented in Figure 3 and Figure 4 respectively for the following discussions. Note that the state machine for TCP-v is the same as that of default TCP, and the interface between pTCP and TCP-v is presented in Section 4.2. We assume the number of interfaces to be nIF at both ends.

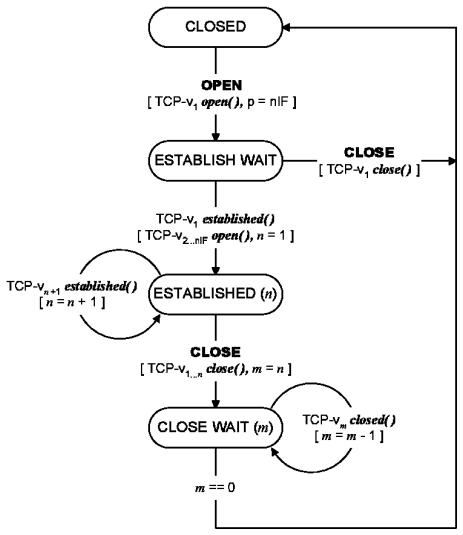


Figure 3: pTCP State Machine

- *Establishment:*

  When an active open is issued by the client application, a pTCP socket with a transmission control block (TCB) similar to TCP's TCB, but with the additional state variables introduced earlier in Section 4.1, is created. After the pTCP socket is created, pTCP creates one TCP-v TCB and issues the $open()$ call to it. When the TCP-v SYN packet is sent out, pTCP sets nIF in the nTx field and the corresponding IP addresses in $ipTx$, and appends additional pTCP connection management header information to the packet. When the pTCP at the server end receives the *passive* open, it checks to see if it is willing to support nIF TCP-v pipes.[3] Assuming that the receiver can support the required number of pipes, it creates the first TCP-v TCB, issues the passive open to it, and in the process takes it to the **SYN_RCVD** state [22]. When the $SYN + ACK$ is sent out by the first TCP-v at the server end, the destination IP address is appropriately set based on the information received in the first SYN, and the source address reflects the local host interface the first TCP-v pipe is bound to. The $SYN + ACK$ message carries nIF in the nRx field that the server has agreed to support, and the corresponding IP addresses in $ipRx$.

  When the client pTCP receives the $SYN + ACK$, it creates the remaining $nIF - 1$ TCP-v TCBs and issues $open()$ calls to each of them. Also, the first TCP-v pipe at this stage enters the **ESTABLISHED** state after sending back an ACK to the server. pTCP thus goes into the **ESTABLISHED (1)** state and can start accepting data from the application. *Hence, even if some of the pipes are experiencing connection setup problems, pTCP will still ensure data flow between the client and the server.*

  The source IP address of each of the outgoing SYNs is set to the local interface the TCP-v pipe is bound to. The destination address is set to one of the addresses in $ipRx$ in the $SYN + ACK$ sent from the server. When the first TCP-v

[3]There might be several reasons including memory or processor limitations, security considerations, etc., because of which the receiving host might desire to limit the number of pipes.

RED HAT 1012
RED HAT V. CAS

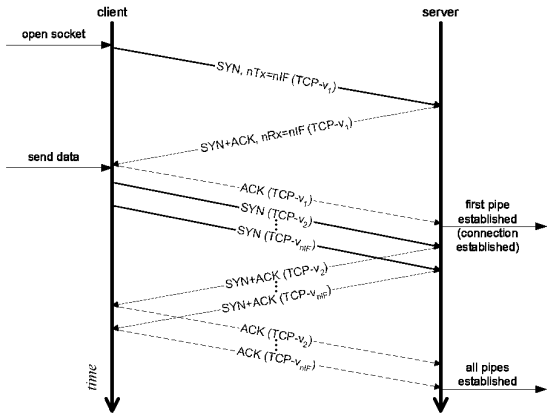


Figure 4: Connection Establishment Handshake

pipe at the server receives the ACK, it enters the **ESTABLISHED** state and can thus participate in the data exchange with the client. The pTCP at the server end also enters the **ESTABLISHED (1)** state. When the server receives the SYN messages from each of the remaining $nIF - 1$ TCP-v pipes, it creates the corresponding TCP-v TCBs and assigns the respective SYNs to the TCBs, taking each of them to the **SYN_RCVD** state. From there on, the exchange of information between each server TCB and the corresponding client TCB is similar to that of TCP.

As and when each of the individual TCP-v pipes enter the **ESTABLISHED** state, they issue the $established()$ call to pTCP making pTCP move down the state machine shown in Figure 3. Finally, when all the individual pipes enter the **ESTABLISHED** state, pTCP enters the **ESTABLISHED (nIF)** state.

- *Termination:*

  The teardown of a pTCP connection is relatively simpler than the connection establishment. When an application closes the connection, pTCP uses the $close()$ interface to make the individual TCP-v pipes close. Each pipe closes using TCP's regular closing handshake. When a TCP-v pipe enters the **CLOSED** state in its state machine, it invokes the $closed()$ callback to pTCP. For every $closed()$ message pTCP receives, it moves down the pTCP state machine. Upon successful completion of all TCP-v pipes, pTCP enters the **CLOSED** state of Figure 3 and confirms the close to the application layer.

4.5 Congestion Control and Flow Control

pTCP by itself does not perform any congestion control. The individual TCP-v pipes are solely responsible for controlling the amount of data transferred through each pipe. On the other hand, flow control in pTCP is performed at the pTCP layer. While the primary reason is the fact that pTCP has control over the receive buffer, it also helps in better utilization of the buffer across the multiple pipes. For example, in the case of the simple application layer approach, irrespective of the bandwidth-delay product (BDP) of the individual TCP pipes, each pipe would have a constant buffer (of 64KB by default). This will result in wastage of buffer space for pipes with smaller BDPs and wastage of capacity for pipes with larger BDPs. However, in pTCP the buffer space will be shared by the individual pipes based on their respective BDPs. Note that this property can also be achieved using some approaches proposed in related work [14].

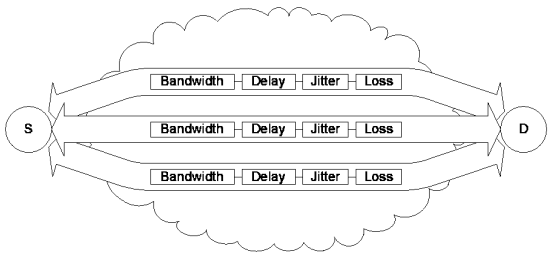


Figure 5: Network Topology

The buffer space (both send and receive) available at the pTCP layer is assumed to be $n * B$, where n is the number of TCP-v pipes, and B is the default TCP buffer size. Every segment that belongs to a pTCP connection always carries the available space in the pTCP receive buffer, irrespective of which pipe it belongs to. The pTCP sender keeps track of the number of outstanding bytes for the connection, and ensures that the receive buffer never overflows. Although all individual TCP-v pipes see the same available buffer space and hence can contend simultaneously for that space (provided there is space in their congestion windows), since pTCP has control over all data transmissions, it prevents any excess data from being transmitted. For example, consider a scenario in which the receiver has advertised a window size of 1000 bytes. Assuming that there exist three TCP-v pipes at the sender and each of them has 1000 bytes space left in the congestion window, each of the pipes will attempt to transmit 1000 bytes worth of data. However, except for the first pipe that succeeds in transmitting the 1000 bytes, the other pipes would have a `NO_DATA` value returned for their $send()$ calls since pTCP would be aware of the global situation.

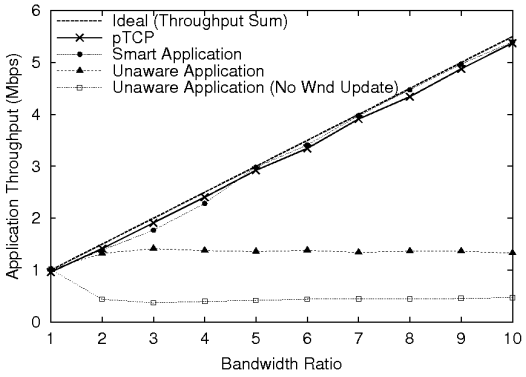
We have thus far described the key components of the pTCP protocol. In the next section, we present performance evaluation results for the pTCP protocol comparing it with the performance of simple and sophisticated application layer techniques.

5. PERFORMANCE EVALUATION

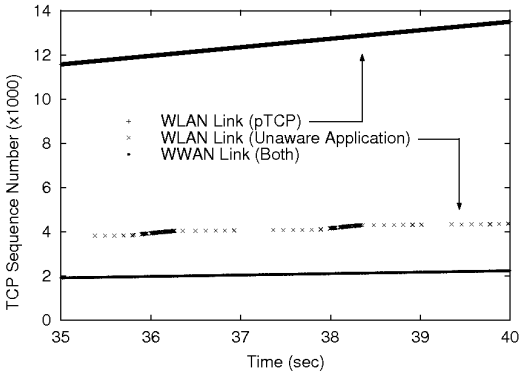
5.1 Simulation Model

We use the *ns2* [15] network simulator and the generic topology shown in Figure 5 for all our simulations. We emulate the end-to-end path from the multi-homed mobile host (S) to the destination (D) through our custom bandwidth, delay, jitter, and loss modules added to the link object in *ns2*. Since we assume the bottleneck to be in the wireless link, we do not explicitly use a sophisticated backbone topology in the simulations. However, we introduce variations in the delay and jitter modules to capture the variations that connections would be expect to observe. We use three primary types of links in most of our simulations: (i) links with bandwidths ranging from 500Kbps to 5Mbps and a round-trip time of 100ms - representative of a connection through a WLAN, (ii) links with a bandwidth of 2Mbps and a round-trip time of 400ms - representative of a connection through a WWAN pico-cell, and (iii) links with a bandwidth of 500Kbps and a round-trip time of 400ms - representative of a connection through a WWAN macro-cell. Packet loss rates from 0.001% to 1% are used in the simulations, and we identify the specific link characteristics used in simulations as we present the different results. We use the TCP-SACK implementation for the transport layer by default. We use two application layer techniques in our comparisons: an "unaware" approach identical to

RED HAT 1012
RED HAT V. CAS



(a) Throughput vs. Bandwidth Ratio



(b) Sequence Number Progression (Ratio = 5)

Figure 6: Scalability with Rate Differential

the one described in Section 2, and a "smart" approach that uses a striping ratio based on the average bandwidths of the different links. pTCP is implemented as a wrapper around TCP-SACK as explained in Section 4. We also present the "ideal" performance of bandwidth aggregation by using multiple applications (one application and one socket for each link) and summing the respective throughputs. We explicitly introduce CBR traffic over UDP as the background traffic when necessary. All packet sizes are set to 1KB. Simulations are run for a period of 600 seconds, and are averaged over 10 samples when randomness is introduced.

We measure throughput (both aggregate and instantaneous) as the metric in our comparisons. We also present the TCP sequence number progression of a connection where appropriate. We present the following results in the rest of the section: (i) scalability with respect to rate differential, (ii) scalability with respect to the number of interfaces, (iii) resilience to rate fluctuations, (iv) resilience to blackouts, and (v) co-existence of different congestion control schemes.

5.2 Simulation Results

5.2.1 Rate Differential

In this section, we use the topology in Figure 5 with two active links between the source and the destination. We fix the bandwidth of one of the links to 500Kbps, and increase the bandwidth of the other link from 500Kbps to 5Mbps in increments of 500Kbps. The round-trip time is fixed as 400ms for the first link, and 100ms for

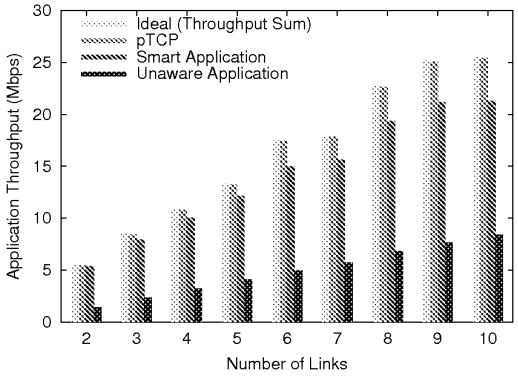


Figure 7: Scalability with Multiple Links

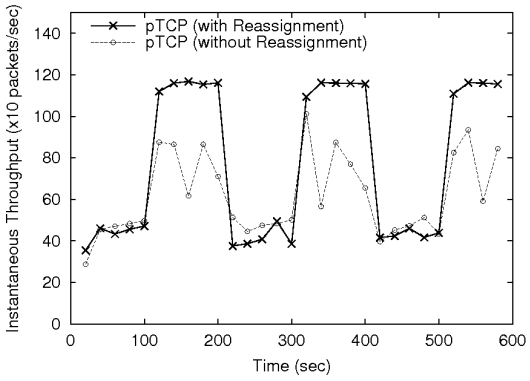
the second. While the first link is representative of a WWAN link, the second link is representative of a WLAN link. For each value of the bandwidth for the second link, we monitor the throughput performance of pTCP, smart application, and unaware application. For reference, we also plot the ideal aggregate throughput curve. We present the aggregate throughput results in Figure 6(a). The x-axis value represents the ratio of the bandwidth of the second link to that of the first. It can be observed that while pTCP and the smart application achieve near ideal performance, the unaware approach performs significantly worse and exhibits a non-increasing aggregate throughput beyond a ratio of 2. We also simulate the scenario when the window update from the TCP receiver is lost. The sending TCP of the faster pipe thus enters persist mode which in turns causes the slower pipe to do the same. As shown in the figure, the result is that the aggregate throughput is lower than that of the slowest pipe as explained in Section 2.

The non-performance of the unaware application, while explained in Section 2, is further illustrated by the results presented in Figure 6(b), where the sequence number progression is shown during a small time window for two pipes with bandwidth ratio of 5. For reference, we also present the case for pTCP. It can be observed that in the unaware application, the head-of-line blocking at the receiver due to the slower pipe stalls the faster pipe. The faster pipe in Figure 6(b) thus exhibits distinct idle periods (e.g. 36.5s to 38s), that results in the degraded performance of the unaware application. In contrast, the results for pTCP for the same scenario exhibits a smooth flow of transmissions through the faster pipe. This is attributed to the congestion window based assignment strategy adopted by pTCP.

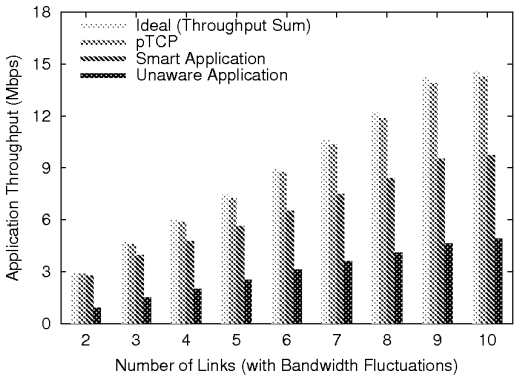
5.2.2 Number of Links

For results in this section, we use the topology in Figure 5, and consider the performance of pTCP and the other approaches as the number of links is increased from 2 to 10. For a scenario with i links, we fix the bandwidth of the first link to 500Kbps, the bandwidth of the second link to 5Mbps, and the bandwidths of the remaining links to values between 500Kbps and 5Mbps randomly chosen. Figure 7 presents the aggregate throughput enjoyed by the application using different striping techniques as the number of links is increased. Since randomness is introduced in the form of the variable link bandwidth assignment, we use averages over 10 scenarios for this result. It can be observed that the performance of pTCP scales well with increasing number of links, while the performance of the unaware application does not. Note that the throughput degradation observed in the smart application as the number

RED HAT 1012
RED HAT V. CAS



(a) Effect of Dynamic Reassignment



(b) Throughput vs. Number of Links

Figure 8: Impact of Fluctuations

of links increases is in fact due to the particular implementation used. Since the smart application always stripes data across multiple pipes based on the bandwidth ratio of the underlying links (the minimum striping unit is one packet), any short-term unfairness exhibited in individual pipes will decrease the effectiveness of the algorithm. As the number of links increases, such short-term unfairness occur more frequently (the periodic TCP probing losses on each link contribute to the unfairness), thus degrading the aggregate throughput enjoyed by the smart application. While we acknowledge that a more sophisticated implementation of the smart application could conceivably be used to solve the problem, its performance will degrade further due to other reasons as shown in the next sections.

5.2.3 *Rate Fluctuations*

Since the characteristics of wireless links exhibit high variances in bandwidths and delays, in this section, we investigate the performance of pTCP in the presence of fluctuations in the available capacity of individual pipes. We show that pTCP's dynamic reassignment design element is effective in addressing the capacity fluctuations, and compare the performance of pTCP with other approaches.

We first consider the topology in Figure 5 with two links. One of the links is fixed at a bandwidth of 5Mbps and a round-trip time of 100ms, while the bandwidth of the other link is initially set to 2Mbps and a round-trip time of 400ms. A *square-wave* CBR flow with a period of t and an amplitude of 1.5Mbps is used as background traffic, to cause the fluctuation on the second link. Hence, the available bandwidth for the second link fluctuates between 0.5Mbps and 2Mbps, with a period of t seconds. While the fluctuations are caused by varying the CBR background load on the appropriate links, they are representative of both congestion based rate changes, and wireless channel condition based rate changes. The result shown in Figure 8(a) is obtained using $t = 200$ seconds. We use the number of packets delivered to the application (over a 20-second time-window) as the instantaneous throughput and compare the performance of pTCP with and without dynamic reassignment. It is clear from the figure that with dynamic reassignment pTCP is able to perform well under bandwidth fluctuations.

In Figure 8(b) we compare the performance of different striping techniques using the multi-link topology used in Section 5.2.2. However, the bandwidth of each link now randomly fluctuates (every 1 second) between 20% and 100% of the normal value. For example, for a link with capacity of 5Mbps, the available bandwidth that the application can use randomly fluctuates between 1Mbps and 5Mbps throughout the simulation. The average bandwidth is thus 60% of the link capacity. We observe that even under such a dynamic environment, the performance of pTCP still closely follows that of the ideal performance. However, this is not the case for the smart and unaware applications. The scenario is appropriate for demonstrating the inefficacy of even a smart application layer approach that does some rate estimation. Its performance will suffer as long as its rate estimation is coarser than the actual fluctuations. Although the smart application can eventually adapt to the changed rate, it cannot do anything about packets already in the pipe that has now slowed down.

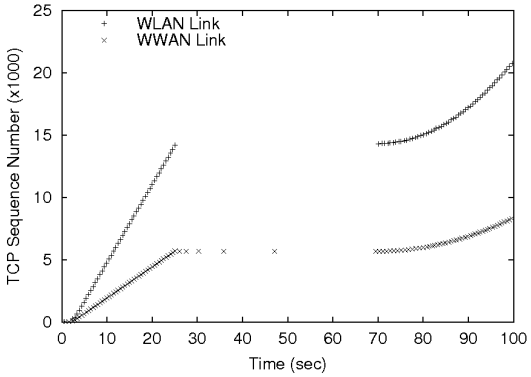
5.2.4 *Blackouts*

In this section, we show the impact of blackouts on the performance of pTCP and application striping techniques. We use the same two link topology used in the previous sections, but introduce a long (25 seconds) blackout between 25s and 50s when the available bandwidth on the WWAN link (2Mbps) decreases to zero. All packets transmitted through the WWAN link during the blackout period are dropped. Figure 9(a) shows the sequence number progression for the smart application on both the WLAN and WWAN pipes. While it is obvious that the WWAN link stops sending data during the blackout period, because of the head-of-line blocking problem described in Section 2, even the WLAN link stalls for most of the blackout period, resulting in the aggregate connection coming to a standstill. On the other hand, as seen in Figure 9(b), in pTCP, although the WWAN link stalls during the blackout period, the WLAN link continues to progress after a minor stall. This is possible because of the redundant striping policy in pTCP that reassigns even the first segment within the congestion window of the WWAN pipe to the WLAN pipe and prevents the latter from experiencing head-of-line blocking.

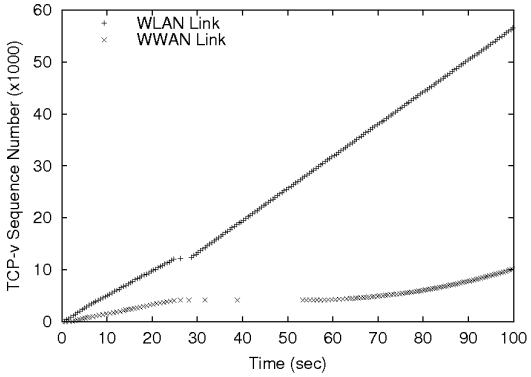
5.2.5 *Different Congestion Control Schemes*

In this section, we demonstrate the ability of the pTCP protocol to use two different congestion control schemes within the same aggregate connection. We consider the two link topology again with bandwidths of 5Mbps (WLAN) and 2Mbps (WWAN), and round-trip times of 100ms and 400ms respectively. A loss module is inserted on the WWAN link. The module inserts losses at packet error rates ranging from 0.001% to 1%. We consider the performance of pTCP when the WLAN link uses regular TCP, and the WWAN link uses TCP-ELN [3]. TCP-ELN receives explicit loss notification from the underlying link layer when a packet is dropped due to random wireless loss, and does not react to such losses. Note

RED HAT 1012
RED HAT V. CAS



(a) Smart Application during Blackouts



(b) pTCP during Blackouts

Figure 9: Impact of Blackouts

that it is not the sophistication of the transport protocol used that is of key importance, but the ability of the pTCP approach to accommodate two different congestion control schemes within the same framework. Figure 10 presents the throughput performance results for pTCP, the smart application using either TCP or TCP-ELN on both links, and the ideal performance (sum of the throughputs of an independent TCP-ELN connection over the WWAN link and an independent TCP connection over the WLAN link). It can be seen that pTCP achieves almost the maximum achievable performance, illustrating the seamless nature in which pTCP allows the two congestion control schemes to co-exist.

6. ISSUES AND DISCUSSION

In this section we discuss some issues with the pTCP design.

- *Congestion Window and Bandwidth-Delay Product:*

  One of the key assumptions made by pTCP when it performs congestion window based striping is that the congestion window is a tight approximation of the available bandwidth-delay product. However, this might not always be true. For example, deep buffering in the network can artificially inflate the congestion window to much larger than the true bandwidth-delay product of a connection. One plausible solution to this problem is to complement the basic congestion control scheme in TCP-v with mechanisms that help estimate the BDP more accurately. For example, if the rate of incoming ACKs for a particular pipe is monitored to keep track of the available rate, once the sending rate of the pipe exceeds the BDP, the rate of incoming ACKs will hit a plateau. pTCP thus can use this as an indication to cap the amount of data assigned to that pipe. Note that since pTCP is implemented as a transport layer approach, it has easier access to the states of TCP than a higher layer approach does. Our current work involves studying this problem more closely.

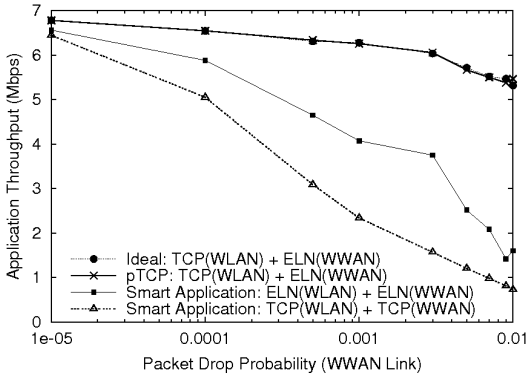


Figure 10: Multiple Congestion Control Schemes in pTCP

- *TCP Friendliness:*

  An important consideration when using multiple interfaces and bandwidth aggregation is how the aggregation plays a role in the TCP friendliness of the connection. Can it happen that in performing the aggregation the end-to-end connection loses its TCP friendly nature? If the assumption made earlier in the paper that the bottlenecks are solely in the wireless domain were true, this would not be an issue. However, if the bottlenecks happen to be in the wired domain, it is possible that multiple TCP-v pipes share a single bottleneck in the wired domain and make the aggregate pTCP connection more aggressive than a single regular TCP connection. There are two possible solutions to this problem: (i) Recent works have proposed schemes to heuristically determine if connections share the same bottleneck by monitoring the packet loss patterns and inter-arrival times [13]. If it can thus be inferred that two TCP-v pipes are sharing the same bottleneck link, approaches like [4] can then be employed to take care that the aggregate behavior of the two TCP-v pipes mimics that of a single regular TCP connection. (ii) Another approach that we have extensively explored in a different context is to use a variant of TCP-Vegas. TCP-Vegas by itself has been shown to be *subservient* to TCP-Reno and TCP-NewReno flows. A more subservient variant of TCP-Vegas (where the congestion window is reset to one upon detection of consistent increase in the history of round-trip times maintained) can be made to use only bandwidth given up by other Reno or NewReno flows. Therefore, in pTCP while one TCP-v pipe would use a Reno or a NewReno congestion control scheme, the other pipes would use a variant of the TCP-Vegas congestion control algorithm, thus lessening the TCP unfriendly effect when multiple TCP-v pipes share the same bottleneck.

- *Backward Compatibility:*

  In this paper, we have made an assumption that both the sender and the receiver are pTCP aware. We believe this assumption to be reasonable under scenarios where mobile

RED HAT 1012
RED HAT V. CAS

users predominantly communicate with proxies that are already mobile-host aware. However, when the mobile hosts communicate with static hosts that can be potentially unaware, the problem can be handled just like it is handled in other protocols such as TCP-SACK or when the timestamp option is used in TCP. When the SYN packet is sent to start the connection, the option is enabled. If the other end replies with the same option, "awareness" is inferred and the respective protocol is used. However, if the other end replies without the option, normal TCP operation resumes. Since pTCP headers and connection establishment handshakes can be implemented through TCP options in the first place, using such "pTCP PERMITTED" techniques seems to be a realistic solution to ensure correct operations when a pTCP aware host communicates with a pTCP unaware host.

- *Handoffs:*

  While we do not consider handoffs explicitly in the paper, note that the handoff experienced by an individual TCP-v connection can be handled in the default manner it would have been handled if it were the only pipe used by the application. In fact, the use of pTCP ensures that even if stalls are caused in one pipe due to handoffs, the other pipes remain unaffected. Moreover, soft handoff can be easily achieved (where the mobile host is connected to multiple access points in the intersection of their coverage areas) using pTCP over the two pipes established to the two access points during handoffs.

- *Complexity:*

  There are two sources of complexity in pTCP that can cause potential problems: (i) The creation and maintenance of multiple TCB states can be a drain on the end-host's resources. pTCP's connection establishment phase addresses this problem by allowing the end-host to accept requests for an aggregate connection by specifying a limit on the number of pipes that it can support. (ii) The overheads incurred by the buffer management at pTCP and individual TCP-v pipes. However, in pTCP the manipulation of the socket buffer incurs the same overhead as the buffer management mechanisms in a regular TCP socket. The only additional overheads occur when packets are unbound following a congestion window reduction. In this case, the unbound packets need to be re-inserted into the unsent list sorted according to sequence numbers. We are currently investigating efficient data structures that can reduce the overheads incurred when the reinsertion is performed.

7. RELATED WORK

We classify related work based on whether the proposed approaches to achieve bandwidth aggregation are performed at the application layer, transport layer, or link layer.

- *Application Layer Techniques:*

  Several approaches have been proposed to use multiple TCP connections in parallel to provide higher throughput to the application. For example, in [17] the authors develop the PSockets library used to stripe data over multiple TCP sockets for a better utilization of the network bandwidth, while avoiding the time-consuming process of manually tuning the TCP buffer size. In [2] the authors develop a new application called XFTP that uses multiple TCP connections to overcome the limitation of TCP window size in long-fat links such as satellite links. Similarly, in [9] an extension of the FTP protocol called GridFTP is developed for bulk data transfer where parallel TCP connections are used to increase the throughput in a bottleneck link. In [7], the authors characterize and substantiate the performance improvement of an aggregate connection that uses parallel TCP connections over the same path. Note that this class of related work deals with using multiple TCP connections over the same path. We discuss in Section 2 the pitfalls of such approaches when used in the context of multi-homed mobile hosts.

- *Transport Layer Techniques:*

  The Stream Control Transmission Protocol (SCTP) is a reliable transport protocol that was designed for the transport of message-based signaling information across IP-based networks [20]. One salient feature of SCTP is the support for multi-streaming and multi-homing. A SCTP connection can consist of multiple data streams across one or multiple interfaces. SCTP provides reliable in-sequence delivery within each data stream, *but it does not provide a total ordering across the data streams.* Although the head-of-line blocking among different data streams is thus avoided in SCTP, it cannot provide bandwidth aggregation as pTCP does. If a SCTP user is to stripe data across multiple data streams, it must handle packet resequencing itself. Another distinct difference between SCTP and pTCP lies in the congestion control mechanism. Multiple data streams within the same SCTP connection are subject to one common flow and congestion control mechanism. In the context of multi-homed mobile hosts where different data streams traverse through vastly differing wireless links and heterogeneous access networks, such design unnecessary leads to bandwidth underutilization for the aggregate connection. The Reliable Multiplexing Transport Protocol (RMTP) is a rate-based transport layer approach that is specifically designed to aggregate bandwidths on multi-homed mobile hosts [10]. Although RMTP targets the same scenario as pTCP does, it differs from pTCP in several ways: (i) RMTP performs explicit bandwidth based striping. The available bandwidth of the underlying pipe is estimated by periodically sending packet-pair probes. The effectiveness of RMTP thus greatly depends on the accuracy of the bandwidth estimation. However, the bandwidth probing rate limits how fast it can detect and adapt to bandwidth fluctuations. When bandwidth fluctuations occur at a time-scale smaller than the bandwidth probing period, RMTP will exhibit the same problem as the smart application does shown in Section 5. (ii) The RMTP design does not explicitly address the interaction between component pipes of the aggregate connection as pTCP does through delayed binding, dynamic reassignment and redundant striping. We show in earlier sections that these design components play an important role in achieving effective bandwidth aggregation on multi-homed mobile hosts. (iii) Finally, RMTP does not provide interfaces allowing the flexible inclusion of different congestion control mechanisms optimized for different wireless links.

- *Link Layer Techniques:*

  As mentioned in Section 1, conventional link layer striping techniques do not perform well in the context of multi-homed mobile hosts, where the multiple interfaces are more likely to belong to different network domains altogether. An ideal striping algorithm not only has to deal with a highly dy-

RED HAT 1012
RED HAT V. CAS

namic and vastly differing set of wireless links, but also has to address fluctuations in capacity caused by the end-to-end multi-hop nature of the paths. In [1], the authors propose a "channel" striping algorithm where the channel is defined as a logical FIFO path at any protocol layer. The authors show that the striping (load sharing) algorithm is in fact the reverse of the fair queueing algorithm. By reversing the direction of packet flow in the fair queueing algorithm, the sender can achieve optimal load sharing across channels of different capacities. If the receiver is running the same fair queueing algorithm used at the sender and no packets are lost, in-sequence delivery can be achieved. Although this algorithm is presented as a generic approach for striping over any logical channel including the transport layer, due to the nature of the fair queueing algorithm *the capacity of each channel must be known a priori at both ends*. Moreover, the states of the algorithms at both ends must be synchronized to ensure in-sequence delivery. However, packet losses cause loss of synchronization between the sender and the receiver. Hence the algorithm has to periodically insert "marker" packets into the channels to achieve resynchronization. Nonetheless, in wireless environments with high loss rates, even the marker packets may get lost, resulting in degrading the performance of the striping algorithm and potentially delivering packets out-of-order to the application. Fluctuations in channel capacity further limit the applicability of this approach in the targeted environment.

8. CONCLUSIONS

In this paper, we consider the problem of using multiple interfaces on a mobile host to provide aggregate bandwidths to applications. Since the multiple interfaces can potentially and will most likely belong to different wireless network domains, link layer striping schemes cannot be use to achieve bandwidth aggregation. At the same time, we show that application layer techniques using default TCP sockets do not scale well when the link characteristics are different and fluctuating. In this context, we propose a transport layer approach called *pTCP* that achieves bandwidth aggregation using a combination of mechanisms including: (i) decoupled congestion control and reliability, (ii) congestion window based striping, (iii) dynamic window reassignment, (iv) redundant striping to handle blackouts, and (v) support for different congestion control schemes to co-exist within a single transport layer framework. We show through simulations that pTCP achieves bandwidth aggregation efficiently under a variety of network conditions.

9. REFERENCES

[1] H. Adiseshu, G. Parulkar, and G. Varghese. A reliable and scalable striping protocol. In *Proceedings of ACM SIGCOMM*, Palo Alto, CA USA, Aug. 1996.

[2] M. Allman, H. Kruse, and S. Ostermann. An application-level solution to TCP's satellite inefficiencies. In *Proceedings of Workshop on Satellite-Based Information Services (WOSBIS)*, Rye, NY USA, Nov. 1996.

[3] H. Balakrishnan, V. Padmanabhan, S. Seshana, and R. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, Dec. 1997.

[4] H. Balakrishnan, H. Rahul, and S. Seshan. An integrated congestion management architecture for Internet host. In *Proceedings of ACM SIGCOMM*, Boston, MA USA, Sept. 1999.

[5] H. Balakrishnan, S. Seshan, and R. Katz. Improving reliable transport and handoff performance in cellular wireless networks. *ACM Wireless Networks*, 1(4):469–481, Dec. 1995.

[6] J. Duncanson. Inverse multiplexing. *IEEE Communications Magazine*, 32(4):34–41, Apr. 1994.

[7] T. Hacker and B. Athey. The end-to-end performance effects of parallel TCP sockets on a lossy wide-area network. In *Proceedings of IEEE IPDPS*, Fort Lauderdale, FL USA, Apr. 2002.

[8] T. Henderson and R. Katz. Transport protocols for Internet-compatible satellite networks. *IEEE Journal on Selected Areas in Communications (JSAC)*, 17(2):345–359, Feb. 1999.

[9] J. Lee, D. Gunter, B. Tierney, B. Allcock, J. Bester, J. Bresnahan, and S. Tuecke. Applied techniques for high bandwidth data transfers across wide area networks. In *Proceedings of Computers in High Energy Physics (CHEP)*, Beijing, China, Sept. 2001.

[10] L. Magalhaes and R. Kravets. Transport level mechanisms for bandwidth aggregation on mobile hosts. In *Proceedings of IEEE ICNP*, Riverside, CA USA, Nov. 2001.

[11] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP selective acknowledgement options. IETF RFC 2018, Oct. 1996.

[12] J. Nagle. Congestion control in IP/TCP internetworks. In *IETF RFC 896*, Jan. 1984.

[13] D. Rubenstein, J. Kurose, and D. Towsley. Detecting shared congestion of flows via end-to-end measurement. In *Proceedings of ACM SIGMETRICS*, Santa Clara, CA USA, June 2000.

[14] J. Semke, J. Mahdavi, and M. Mathis. Automatic TCP buffer tuning. In *Proceedings of ACM SIGCOMM*, Vancouver, Canada, Sept. 1998.

[15] The Network Simulator. *ns-2*. http://www.isi.edu/nsnam/ns.

[16] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan. WTCP: A reliable transport protocol for wireless wide-area networks. In *Proceedings of ACM MOBICOM*, Seattle, WA USA, Aug. 1999.

[17] H. Sivakumar, S. Bailey, and R. Grossman. PSockets: The case for application-level network striping for data intensive applications using high speed wide area networks. In *Proceedings of IEEE Supercomputing (SC)*, Dallas, TX USA, Nov. 2000.

[18] A. Snoeren. Adaptive inverse multiplexing for wide-area wireless networks. In *Proceedings of IEEE GLOBECOM*, Rio de Janeireo, Brazil, Dec. 1999.

[19] M. Stemm and R. Katz. Vertical handoffs in wireless overlay networks. *Mobile Networks and Applications*, 3(4):335–350, 1998.

[20] R. Stewart et al. Stream control transmission protocol. IETF RFC 2960, Oct. 2000.

[21] C. B. Traw and J. Smith. Striping within the network subsystem. *IEEE Network Magazine*, 9(4):22–32, July 1995.

[22] G. R. Wright and W. R. Stevens. *TCP/IP Illustrated, Volume 2*. Addison-Wesley Publishing Company, Reading, MA USA, Oct. 1997.

RED HAT 1012
RED HAT V. CAS

The Click Modular Router

EDDIE KOHLER, ROBERT MORRIS, BENJIE CHEN, JOHN JANNOTTI,
and M. FRANS KAASHOEK
Laboratory for Computer Science, MIT

Click is a new software architecture for building flexible and configurable routers. A Click router is assembled from packet processing modules called *elements*. Individual elements implement simple router functions like packet classification, queueing, scheduling, and interfacing with network devices. A router configuration is a directed graph with elements at the vertices; packets flow along the edges of the graph. Several features make individual elements more powerful and complex configurations easier to write, including *pull connections*, which model packet flow driven by transmitting hardware devices, and *flow-based router context*, which helps an element locate other interesting elements.

Click configurations are modular and easy to extend. A standards-compliant Click IP router has sixteen elements on its forwarding path; some of its elements are also useful in Ethernet switches and IP tunneling configurations. Extending the IP router to support dropping policies, fairness among flows, or Differentiated Services simply requires adding a couple elements at the right place. On conventional PC hardware, the Click IP router achieves a maximum loss-free forwarding rate of 333,000 64-byte packets per second, demonstrating that Click's modular and flexible architecture is compatible with good performance.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Packet-switching networks*; C.2.6 [**Computer-Communication Networks**]: Internetworking—*Routers*; D.2.11 [**Software Engineering**]: Software Architectures—*Domain-specific architectures*

General Terms: Design, Management, Performance

Additional Key Words and Phrases: Routers, component systems, software router performance

An article describing a previous version of this system was published in *Operating Systems Review* **34**(5) (*Proceedings of the 17th Symposium on Operating Systems Principles*), pp 217–231, December 1999.

This research was supported by a National Science Foundation (NSF) Young Investigator Award and the Defense Advanced Research Projects Agency (DARPA) and Rome Laboratory under agreement number F30602-97-2-0288. In addition, Eddie Kohler was supported by a National Science Foundation Graduate Research Fellowship.

Address: MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139; email: {eddietwo; rtm; benjie; jj; kaashoek}@lcs.mit.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works, requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept, ACM Inc., 1515 Broadway, New York, NY 10036 USA, fax +1 (212) 869-0481, or `permissions@acm.org`.

RED HAT 1012
RED HAT V. CAS

1. INTRODUCTION

Routers are increasingly expected to do more than route packets. Boundary routers, which lie on the borders between organizations, must often prioritize traffic, translate network addresses, tunnel and filter packets, and act as firewalls, among other things. Furthermore, fundamental properties like packet dropping policies are still under active research [Floyd and Jacobson 1993; Lakshman et al. 1996; Cisco Corporation 1999], and initiatives like Differentiated Services [Blake et al. 1998] bring the need for flexibility close to the core of the Internet.

Unfortunately, most routers have closed, static, and inflexible designs. Network administrators may be able to turn router functions on or off, but they cannot easily specify or even identify the interactions of different functions. Furthermore, it is difficult for network administrators and third party software vendors to extend a router with new functions. Extensions require access to software interfaces in the forwarding path, but these often don't exist, don't exist at the right point, or aren't published.

This paper presents Click, a flexible, modular software architecture for creating routers. Click routers are built from fine-grained components; this supports fine-grained extensions throughout the forwarding path. The components are packet processing modules called *elements*. The basic element interface is narrow, consisting mostly of functions for initialization and packet handoff, but elements can extend it to support other functions (such as reporting queue lengths). To build a router configuration, the user chooses a collection of elements and connects them into a directed graph. The graph's edges, which are called *connections*, represent possible paths for packet handoff. To extend a configuration, the user can write new elements or compose existing elements in new ways, much as UNIX allows one to build complex applications directly or by composing simpler ones using pipes.

Several aspects of the Click architecture were directly inspired by properties of routers. First, packet handoff along a connection may be initiated by either the source end (*push processing*) or the destination end (*pull processing*). This cleanly models most router packet flow patterns, and pull processing makes it possible to write composable packet schedulers. Second, the *flow-based router context* mechanism lets an element automatically locate other elements on which it depends; it is based on the observation that relevant elements are often connected by the flow of packets. These features make individual elements more powerful and configurations easier to write. For example, in Section 4.2, we present an element that can implement four variants of the Random Early Detection dropping policy [Floyd and Jacobson 1993] depending on its context in the router: RED, RED over multiple queues, weighted RED, or drop-from-front RED. This would be difficult or impossible to achieve in previous modular networking systems.

We have implemented Click on general-purpose PC hardware as an extension to the Linux kernel. A modular Click IP router configuration has a maximum loss-free forwarding rate of 333,000 64-byte packets per second on a 700 MHz Pentium III. This is roughly four times the rate for a standard Linux router on

RED HAT 1012
RED HAT V. CAS

the same hardware, mostly due to device handling improvements inspired by previous work [Mogul and Ramakrishnan 1997; Druschel et al. 1994]. It is also slightly more than the rate for a modified Linux router that uses our network device extensions, demonstrating that Click's modularity and flexibility comes at an acceptable cost.

In the rest of this paper, we describe Click's architecture in detail, including the language used to describe configurations (Section 2), then present a standards-compliant Click IP router (Section 3) and some router extensions (Section 4). After summarizing Click's kernel environment (Section 5) and evaluating its performance (Section 6), we describe related work (Section 7) and summarize our conclusions (Section 8).

2. ARCHITECTURE

A Click *element* represents a unit of router processing. An element represents a conceptually simple computation, such as decrementing an IP packet's time-to-live field, rather than a large, complex computation, such as IP routing. A Click router configuration is a directed graph with elements at the vertices. An edge, or *connection*, between two elements represents a possible path for packet transfer. Every action performed by a Click router's software is encapsulated in an element, from device handling and routing table lookups to queueing and counting packets. The user determines what a Click router does by choosing the elements to be used and the connections among them. Inside a running router, each element is a C++ object that may maintain private state. Connections are represented as pointers to element objects, and passing a packet along a connection is implemented as a single virtual function call.

The most important properties of an element are:

—**Element class.** Each element belongs to one element class. This specifies the code that should be executed when the element processes a packet, as well as the element's initialization procedure and data layout.

—**Ports.** An element can have any number of input and output ports. Every connection goes from an output port on one element to an input port on another. Different ports can have different semantics; for example, second output ports are often used to emit erroneous packets.

—**Configuration string.** The optional configuration string contains additional arguments that are passed to the element at router initialization time. Many element classes use these arguments to set per-element state and fine-tune their behavior.

—**Method interfaces.** Each element supports one or more method interfaces. Every element supports the simple packet-transfer interface, but elements can create and export arbitrary additional interfaces; for example, a queue might export an interface that reports its length. Elements communicate at run time through these interfaces, which can contain both methods and data.

Figure 1 shows how we diagram these properties for a sample element, *Tee(2)*. '*Tee*' is the element class; a *Tee* copies each packet received on its single input port, sending one copy to each output port. Configuration strings are enclosed

RED HAT 1012
RED HAT V. CAS

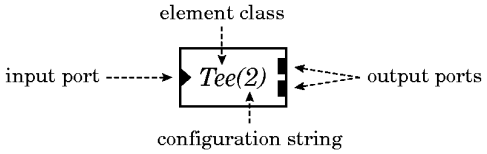


Fig. 1. A sample element. Triangular ports are inputs and rectangular ports are outputs.

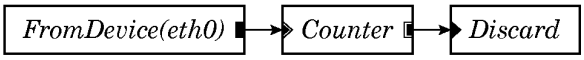


Fig. 2. A router configuration that throws away all packets.

in parentheses; the '2' in '*Tee(2)*' is interpreted by *Tee* as a request for two outputs. Method interfaces are not shown explicitly, as they are implied by the element class. Figure 2 shows several elements connected into a simple router that counts incoming packets, then throws them all away.

2.1 Push and Pull Connections

Click supports two kinds of connections, *push* and *pull*. On a push connection, packets start at the source element and are passed downstream to the destination element. This corresponds to the way packets move through most software routers. On a pull connection, in contrast, the destination element initiates packet transfer: it asks the source element to return a packet, or a null pointer if no packet is available. This is the dual of a push connection. (Clark called pull connections *upcalls* [Clark 1985].) Each of these forms of packet transfer is implemented by one virtual function call.

The processing type of a connection—whether it is push or pull—is determined by the ports at its endpoints. Each port in a running router is either push or pull; connections between two push ports are push, and connections between two pull ports are pull. Connections between a push port and a pull port are illegal. Elements set their ports' types as the router is initialized. They may also create *agnostic ports*, which behave as push when connected to push ports and pull when connected to pull ports. When a router is initialized, the system propagates constraints until every agnostic port has been assigned to either push or pull.[1] In our configuration diagrams, black ports are push and white ports are pull; agnostic ports are shown as push or pull ports with a double outline. Figure 3 shows how push and pull work in a simple router.

Push processing is appropriate when unsolicited packets arrive at a Click router—for example, when packets arrive from a device. The router must handle such packets as they arrive, if only to queue them for later consideration. Pull processing is appropriate when the Click router needs to control the timing of packet processing. For example, a router may transmit a packet only when the transmitting device is ready. In Click, transmitting devices are elements

[1]The simplest way of creating an agnostic port causes each packet handoff to that port to take two virtual function calls rather than one. The first calls a general `push` or `pull` method, which is a wrapper that calls the element's "`agnostic`" method.

RED HAT 1012
RED HAT V. CAS

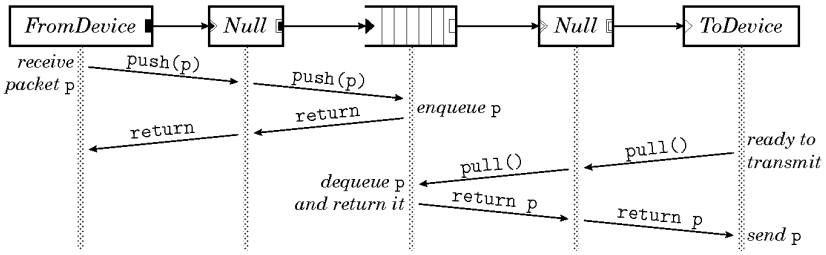


Fig. 3. Push and pull control flow. This diagram shows functions called as a packet moves through a simple router; time moves downwards. The central element is a *Queue*. During the push, control flow moves forward through the element graph starting at the receiving device; during the pull, control flow moves backward through the graph, starting at the transmitting device. The packet p always moves forward.

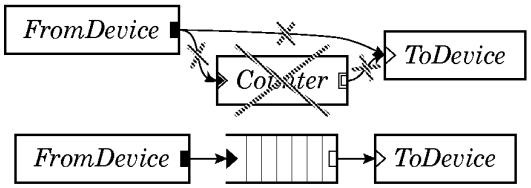


Fig. 4. Some push and pull violations. The top configuration has four errors: (1) *FromDevice*'s push output connects to *ToDevice*'s pull input; (2) more than one connection to *FromDevice*'s push output; (3) more than one connection to *ToDevice*'s pull input; and (4) an agnostic element, *Counter*, in a mixed push/pull context. The bottom configuration, which includes a *Queue*, is legal. In a properly configured router, the port colors on either end of each connection will match.

with one pull input; they therefore initiate packet transfer, and can ask for packets only when they are ready.

Pull processing also models the scheduling decision inherent in choosing the next packet to send. A Click packet scheduler is simply an element with one pull output and multiple pull inputs. Such an element responds to a pull request by choosing one of its inputs, making a pull request to that input, and returning the packet it receives. (If it receives a null pointer, it will generally try another input.) These elements make only local decisions: different scheduling behaviors correspond to different algorithms for choosing an input. Thus, they are easily composable. Section 4.1 discusses this further.

The following properties hold for all correctly configured routers: Push outputs must be connected to push inputs, and pull outputs must be connected to pull inputs. Each agnostic port must be used as push or pull exclusively. Furthermore, if packets arriving on an agnostic input might be emitted immediately on one of that element's agnostic outputs, then both input and output must be used in the same way (either push or pull). Finally, push outputs and pull inputs must be connected exactly once. This ensures that each packet transfer request—either pushing to an output port or pulling from an input port—is along a unique connection. These properties are automatically checked by the system during router initialization. Figure 4 demonstrates some property violations.

RED HAT 1012
RED HAT V. CAS

These properties are designed to catch intuitively invalid configurations. For example, the connection in Figure 4 from *FromDevice* to *ToDevice* is illegal because *FromDevice*'s output is push while *ToDevice*'s input is pull. But this connection is intuitively illegal, since it would mean that *ToDevice* might receive packets when it was not ready to send them. The *Queue* element, which converts from push to pull, also provides the temporary packet storage this configuration requires.

2.2 Packet Storage

Click elements do not have implicit queues on their input and output ports, or the associated performance and complexity costs. Instead, queues in Click are explicit objects, implemented by a separate *Queue* element. This gives the router designer explicit control over an important router property, namely how packets are stored. It also enables valuable configurations that are difficult to arrange otherwise—for example, a single queue feeding multiple devices, or a queue feeding a traffic shaper on the way to a device. Explicit queues necessitate both push and pull connections. A *Queue* has a push input port and a pull output port; the input port responds to pushed packets by enqueueing them, and the output port responds to pull requests by dequeueing packets and returning them.

2.3 CPU Scheduling

Click schedules the router's CPU with a task queue; the router's driver is a loop that processes the task queue one element at a time. (The task queue is currently scheduled with the flexible and lightweight stride scheduling algorithm [Waldspurger and Weihl 1995].) Each task is simply an element that would like special access to CPU time. Thus, the element is Click's unit of CPU scheduling as well as its unit of packet processing. A task can initiate an arbitrary sequence of push and pull requests. However, most elements are never placed on the task queue; they are implicitly scheduled when their `push` or `pull` methods are called. An element should be on the task queue if it frequently initiates push or pull requests without receiving a corresponding request. This includes device elements—for example, *FromDevice* initiates a push request when it receives a packet from a device—as well as elements that ship packets between *Queue*s inside the router configuration.

Another task structure handles timer events. Each element can have any number of active timers, where each timer calls an arbitrary method when it fires.

Click currently runs in a single thread. Thus, any packet transfer method (`push` or `pull`) must return to its caller before another task can begin. The router will continue to process each pushed packet, following it from element to element along a path in the router graph, until it is explicitly stored or dropped (and similarly for pull requests). Therefore, the placement of *Queue*s in a configuration graph determines how CPU scheduling may be performed. For example, if *Queue*s are late in the graph, the router commits to a lot of processing on each input packet before processing the next input packet—specifically, it commits to the processing required to push the packet to a *Queue*.

RED HAT 1012
RED HAT V. CAS

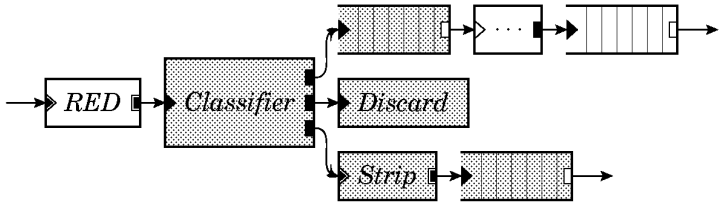


Fig. 5. Flow-based router context. A packet starting at *RED* and stopping at the first *Queue* it encountered might pass through any of the grey elements.

2.4 Flow-Based Router Context

If an element A wants to use a method interface of another element B, it must first locate B. Connections solve this problem for packet transfer, but not for other method interface interactions. Instead, A can refer to B by name (for example, A's configuration string could contain the string "B"), or it can use an automatic mechanism called *flow-based router context*.

Flow-based router context describes where packets starting at a given element might end up after several transfers, or where packets arriving at that element might have originated. This generalizes connections, which specify where a packet might travel in exactly one transfer. In practical terms, elements may ask the system questions such as "If I were to emit a packet on my second output, where might it go?"; the answer is computed using flow-based router context. The element may restrict this question by method interface; when searching for *Queue*s, it might ask "If I were to emit a packet on my second output, which *Queue*s might it encounter?" It may further restrict the answer to the closest method interfaces: "If I were to emit a packet on my second output, and it stopped at the first *Queue* it encountered, where might it stop?" This occupies a useful middle ground between purely local information (connections) and purely global information (the entire router). It can be more robust than naming elements explicitly, since it automatically adapts to changes in the router configuration. It also captures a fundamental router property: if two elements interact, then packets can usually pass from one to the other.

For example, each *Queue* exports its current length using a method interface. Elements such as *RED* (a dropping policy element described further in Section 4.2) are interested in this information, but before finding a *Queue*'s length, *RED* must find the correct *Queue*. It can do so with flow-based router context. Figure 5 shows the router context downstream of *RED*. Every element in the figure is downstream of *RED*, but only the grey elements are found if the search stops at the closest *Queue*s. Thus, *RED*'s flow-based router context search will return the two grey *Queue*s.

Flow-based router context is calculated by a simple data-flow algorithm that is robust in the presence of cycles in the configuration graph. Elements generally ask for flow-based router context once, at router initialization time, and save its results for quick reference as the router runs. Any element that uses flow-based router context must be prepared to handle zero, one, two, or more

RED HAT 1012
RED HAT V. CAS

```
// Declare three elements ...
src :: FromDevice(eth0);
ctr :: Counter;
sink :: Discard;
// ... and connect them together
src -> ctr;
ctr -> sink;

// Alternate definition using syntactic sugar
FromDevice(eth0) -> Counter -> Discard;
```

Fig. 6. Two Click-language definitions for the trivial router of Figure 2.

result elements, possibly by reporting an error if there are too many or too few results. Section 4.2 demonstrates the flexibility benefits of supporting variable numbers of results.

2.5 Language

Click configurations are written in a simple language with two important constructs: *declarations* create elements, and *connections* say how they should be connected. Its syntax is easy enough to learn from an example; Figure 6 uses it to define a trivial router.

The language contains an abstraction mechanism called *compound elements* that lets users define their own element classes. A compound element is a router configuration fragment that behaves like an element class. A user could define a compound element consisting of a *Queue* followed by a *Shaper*, and call the resulting element class *ShapedQueue*; the rest of the configuration could then use *ShapedQueue* as if it were a primitive Click element class. Compound elements can have multiple ports with arbitrary push/pull characteristics. At initialization time, each use of a compound element is compiled into the corresponding collection of simple elements.

The language is wholly declarative: it specifies what elements to create and how they should be connected, not how to process packets procedurally. This makes configurations easy to manipulate—processing a language file loses no information, and there is a canonical language file for any configuration. Router manipulation tools can take advantage of these properties to optimize router configurations off line or prove simple properties about them.

Language extensions are generally implemented through special element classes rather than new syntactic constructs. For example, the language has no special syntax for specifying CPU scheduling priorities. Instead, the user writes this information in the configuration strings of one or more *ScheduleInfo* elements. These elements have no input or output ports, and packets do not pass through them; they are used only at initialization time, for the data in their configuration strings. This design keeps the language clean and simplifies tool maintenance.

2.6 Installing Configurations

There are currently two drivers that can run Click router configurations, a Linux in-kernel driver and a user-level driver that communicates with the net-

RED HAT 1012
RED HAT V. CAS

work using Berkeley packet filters [McCanne and Jacobson 1993] or a similar packet socket mechanism. The user-level driver is most useful for profiling and debugging, while the in-kernel driver is good for production work. The rest of this paper concentrates on the in-kernel driver.

To install a Click router configuration, the user passes a Click-language file to the kernel driver. The driver then parses the file, checks it for errors, initializes every element, and puts the router on line. It breaks the initialization process into stages, allowing cyclic configurations without forcing any particular initialization order. In the early stages, elements set object variables, add and remove ports, and specify whether those ports are push or pull. In later stages, they query flow-based router context, place themselves on the task queue, and attach to Linux kernel structures.

Installing a new configuration normally destroys any old configuration; for instance, any packets stored in old queues are dropped. This starts the new configuration from a predictable empty state. However, Click supports two techniques for changing a configuration without losing information:

—**Handlers.** Each element can easily install any number of *handlers*, which are access points for user interaction. They appear to the user as files in Linux's `/proc` file system; for example, a `count` handler belonging to an element named `c` would be accessible in the file `/proc/click/c/count`. One of `c`'s methods is called when the user reads or writes this file. This lightweight mechanism is most appropriate for modifications local to an element, such as changing a maximum queue length. A Click routing table element, for example, would likely provide `add_route` and `del_route` handlers as access points for user-level routing protocol implementations. Handlers are also useful for exporting statistics and other element information.

—**Hot swapping.** Some configuration changes, such as adding new elements, are more complex than handlers can support. In these cases, the user can write a new configuration file and install it with a hot-swapping option. This will only install the new configuration if it initializes correctly—if there are any errors, the old configuration will continue routing packets without a break. Also, if the new configuration is correct, it will atomically take the old configuration's state before being placed on line; for example, any enqueued packets are moved into the new configuration. This happens only with element classes that explicitly support it, and can be prevented by giving the new elements different names than the old ones.

Finally, element class definitions can be dynamically added to and removed from the Click kernel driver; combined with hot swapping, this makes Click an interesting platform for active networking.

2.7 Element Implementation

Each Click element class corresponds to a subclass of the C++ class `Element`, which has about 20 virtual functions. `Element` provides reasonable default implementations for many of these, so most subclasses must override just six of them or less. Only three virtual functions are used during router operation, namely `push`, `pull`, and `run_scheduled` (used by the task scheduler); the others

RED HAT 1012
RED HAT V. CAS

```
class NullElement: public Element { public:
  NullElement()                        { add_input(); add_output(); }
  const char *class_name() const       { return "Null"; }
  NullElement *clone() const           { return new NullElement; }
  const char *processing() const       { return AGNOSTIC; }
  void push(int port, Packet *p)       { output(0).push(p); }
  Packet *pull(int port)               { return input(0).pull(); }
};
```

Fig. 7. The complete implementation of a do-nothing element: *Null* passes packets from its single input to its single output unchanged.

are used for identification, push and pull specification, configuration, initialization, and statistics.

Subclasses of `Element` are easy to write, so we expect that users will easily write new element classes as needed. In fact, the complete implementation of a simple working element class takes less than 10 lines of code; see Figure 7. Most elements define functions for configuration string parsing and initialization in addition to those in Figure 7, and take about 120 lines of code. When linked with 100 common elements, the Linux kernel driver contains roughly 14,000 lines of core and library source code and 19,000 lines of element source code (not counting comments); this compiles to about 341,000 bytes of i386 instructions, most of which are used only at router initialization time. A simple element's `push` or `pull` function compiles into a few dozen i386 instructions.

2.8 Element Design and Architectural Limitations

A Click user will generally prefer fine-grained elements, which have simple specifications, to coarse-grained elements with more complex specifications. For IP routing, for example, a collection of small elements is preferable to a single element, since the collection of small elements supports arbitrary extensions and modifications via configuration graph manipulation. However, small elements are not appropriate for all problems, since Click's main organizational principle is packet flow. Coarse-grained elements are required when control or data flow doesn't match the flow of packets. For example, complex protocol processing often requires a coarse-grained element; a routing protocol like BGP does not naturally break into parts among which packets flow.

A conventional router contains shared structures that don't participate in packet forwarding, such as routing tables, network statistics, and so forth. In Click, these structures are more naturally incorporated into the packet forwarding path. Routing tables, such as the IP routing table and the ARP cache, are encapsulated by elements that make routing decisions, and statistics are localized inside the elements responsible for collecting them. Of course, these elements can export a method interface so other elements can access the structures.

Several other modular networking systems are built around an abstraction that represents individual network flows [Hutchinson and Peterson 1991; Mosberger and Peterson 1996]. These systems automatically create and destroy modules as network flows are encountered. This is a fast, limited form of con-

RED HAT 1012
RED HAT V. CAS

figuration installation, as each new or deleted flow changes a localized section of the configuration. Hot-swap installation is fast in Click—on a 700 MHz Pentium III, installing a 50-element configuration takes less than a tenth of a second—but not fast enough to support flow creation and deletion. Most of the benefits of a flow-based design can be realized in Click as is; many configurations only require per-flow-class state and CPU scheduling, and elements can cooperate to maintain per-flow private state. Unlike flow-based systems, however, Click cannot schedule the CPU per individual flow.

Flow-based router context is appropriate for many situations where a method interface must be located, but it is not always sufficiently specific. Consider a configuration where bad packets, which occur rarely, are sent to a set of elements for error handling. Upstream elements using flow-based router context will always include the error-handling elements in their search, which might or might not be what the user wanted. Most elements adopt a simple solution: the user can override flow-based router context with a configuration argument when more specificity is required.

The Click language contains no notion of configuration variables. For example, it would be useful to refer to "the Ethernet address of the device handled by that *FromDevice* element" in a configuration string. Currently, the user must copy such an Ethernet address everywhere it is required, which leads to duplication of information. A configuration-variable extension would be simple to add.

3. AN IP ROUTER

This section presents a real Click router configuration: an IP router that forwards unicast packets in nearly full compliance with the standards [Baker 1995; Postel 1981a; Postel 1981b]. Figure 8 shows this configuration for a bridge router with two network interfaces. (The reader may want to refer to Appendix 8, a glossary of Click elements used in the IP router and elsewhere in this paper.) The rest of this section describes the IP router in more detail; Section 4 shows how to extend it by changing its scheduling and queueing behavior, and Section 6 evaluates its performance.

IP forwarding tasks that involve only local information fit naturally into the Click framework. For example, *DecIPTTL* decides if a packet's time-to-live field (TTL) has expired. If the TTL is still valid, *DecIPTTL* decrements it, incrementally updates the packet's checksum, and emits the packet on its first output; if the TTL has expired, *DecIPTTL* emits the packet on its second output (usually connected to an element that generates ICMP errors). These actions depend only on the packet's contents, and do not interact with decisions made elsewhere except as expressed in the packet's path through the element graph. Self-contained elements like *DecIPTTL* can be easily composed. For example, one could connect *DecIPTTL*'s "expired" output to a *Discard* to avoid generating ICMP errors, or insert an element that limited the rate at which errors are generated.

Some forwarding tasks require that information about a packet be calculated in one place and used in another. Click uses *annotations* to carry such information along. An annotation is a piece of information attached to the packet

RED HAT 1012
RED HAT V. CAS

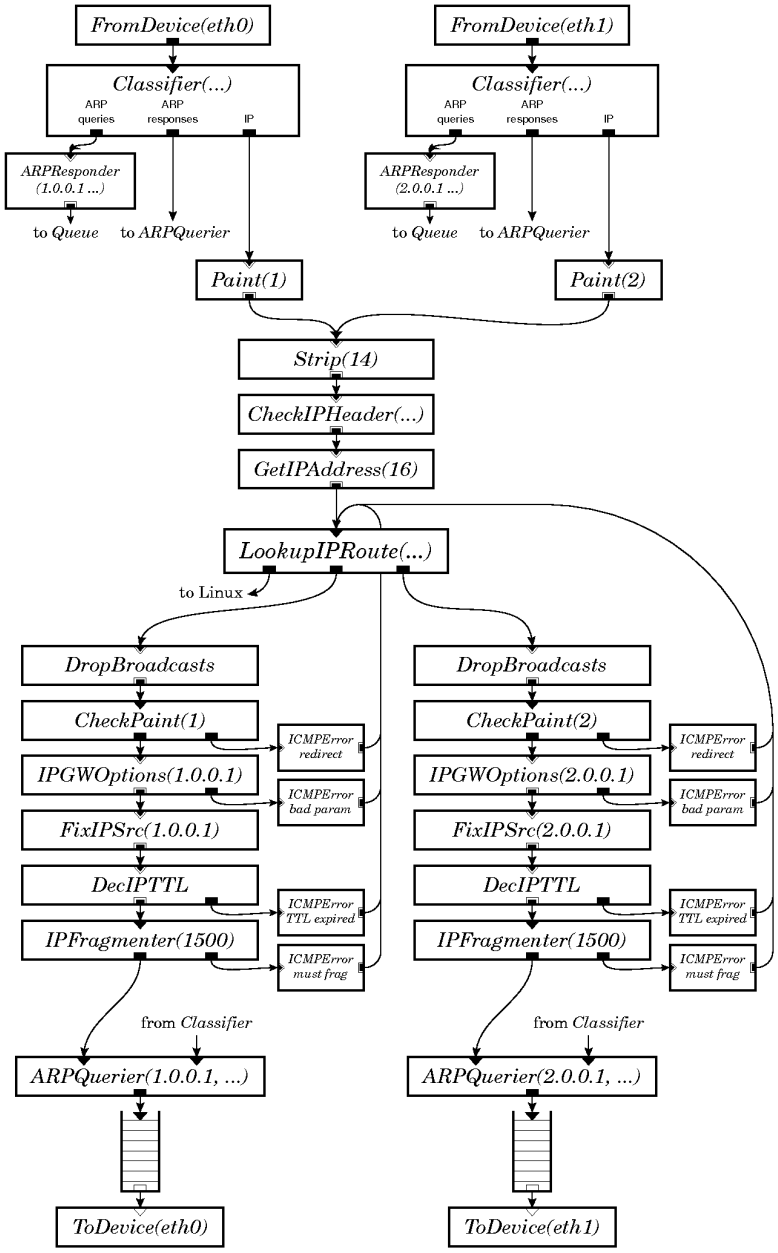


Fig. 8. An IP router configuration.

RED HAT 1012
RED HAT V. CAS

header, but not part of the packet data; the Linux packet structure, `sk_buff`, provides 48 bytes of annotation space. The annotations used in the IP router include:

—**Destination IP address.** Elements that deal with a packet's destination IP address use this annotation rather than the IP header field. This allows elements to modify the destination address—for example, to set it to the next-hop gateway address—without modifying the packet. *GetIPAddress* copies an address from the IP header into the annotation, *LookupIPRoute* replaces the annotation with the next-hop gateway's address, and *ARPQuerier* maps the annotation to the next-hop Ethernet address.

—**Paint.** The *Paint* element marks a packet with an integer "color". *CheckPaint* emits every packet on its first output, and a copy of each packet with a specified color on its second output. The IP router uses color to decide whether a packet is leaving the same interface on which it arrived, and thus should prompt an ICMP redirect.

—**Link-level broadcast flag.** *FromDevice* sets this flag on packets that arrived as link-level broadcasts. The IP router uses *DropBroadcast* to drop such packets if they are about to be forwarded to another interface.

—**ICMP Parameter Problem pointer.** This is set by *IPGWOptions* on erroneous packets to specify the bad IP header byte, and used by *ICMPError* when constructing an error message.

—**Fix IP Source flag.** The IP source address of an ICMP error packet must be the address of the interface on which the error is sent. *ICMPError* can't predict this interface, so it uses a default address and sets the Fix IP Source annotation. After the ICMP packet has been routed towards a particular interface, a *FixIPSrc* on that path will see the flag, insert the correct source address, and recompute the IP checksum.

In a few cases elements require information of an inconveniently global nature. A router usually has a separate IP address on each attached network, and each network usually has a separate IP broadcast address. All of these addresses must be known at multiple points in the Click configuration: *LookupIPRoute* must decide if a packet is destined to the router itself, *CheckIPHeader* must discard a packet with any of the IP broadcast addresses as its source address, *ICMPError* must suppress responses to IP broadcasts, and *IPGWOptions* must recognize any of the router's addresses in an IP Timestamp option. Each of these elements takes the complete list of addresses as part of its configuration string, but ideally they would derive the list automatically (perhaps using flow-based router context).

The IP router conforms to the routing standards both because of the behavior of individual elements and because of their arrangement. In terms of individual elements, the *CheckIPHeader* element checks exactly the properties required by the standards: the IP length fields, the IP source address, and the IP checksum. Also, *IPGWOptions* processes just the Record Route and Timestamp options, since the source route options should be processed only on packets addressed to the router. *IPFragmenter* fragments packets larger than the configured MTU,

RED HAT 1012
RED HAT V. CAS

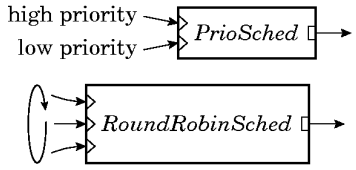


Fig. 9. Some composable packet scheduler elements.

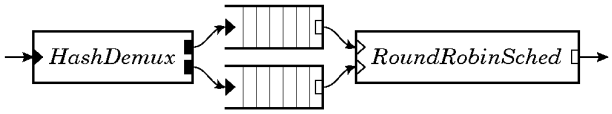


Fig. 10. A virtual queue implementing Stochastic Fairness Queueing.

but sends large, unfragmentable packets to an error output instead. Finally, *ICMPError*, which encapsulates most input packets in an ICMP error message and outputs the result, does not respond to broadcasts, ICMP errors, fragments, and source-routed packets; ICMP errors should not be generated in response to these packets. In terms of element arrangement properties, the standards mandate the placement of *DecIPTTL* after the *LookupIPRoute* routing element; a packet's time-to-live can be decremented only after it is determined that the packet is not destined for the router itself.

4. EXTENSIONS

The Click IP router is modular—it has sixteen elements on its forwarding path—and, therefore, easy to understand and easy to extend. This section demonstrates these benefits of modularity by presenting IP router extensions and other configurations built with the same elements. We show simple extensions supporting scheduling and dropping policies, queueing requirements, and Differentiated Services; an IP tunneling configuration; and one non-IP router, an Ethernet switch. The Click software supports many other extensions not described here, including RFC 2507-compatible IP header compression and decompression, IP security, communication with wireless radios, network address translation, firewalling, and other specialized routing tasks.

4.1 Scheduling

Packet scheduling is a kind of multiplexing: a scheduler decides how a number of packet sources—usually queues—will share a single output channel. A Click packet scheduler is naturally implemented as a pull element with multiple inputs and one output. This element reacts to requests for packets by choosing one of its inputs, pulling a packet from it, and returning that packet. (If the chosen input has no packets ready, the scheduler will usually try other inputs.)

We have implemented three packet scheduler elements: *RoundRobinSched*, *PrioSched*, and *StrideSched*. *RoundRobinSched* pulls from its inputs in round-robin order, returning the first packet it finds (or no packet, if no input has a packet ready). It always starts with the input cyclically following the last successful pull. *PrioSched* is a strict priority scheduler; it always tries its

RED HAT 1012
RED HAT V. CAS

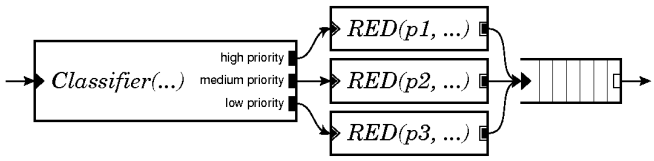


Fig. 11. Weighted RED. The three *RED* elements have different RED parameters, so packets with different priorities are dropped with different probabilities.

first input, then its second, and so forth, returning the first packet it gets. *StrideSched* implements a stride scheduling algorithm [Waldspurger and Weihl 1995] with ticket values specified by the user. Other arbitrary packet scheduling algorithms could be implemented in the same way. *RoundRobinSched* and *PrioSched* are illustrated in Figure 9.

Both *Queue*s and scheduling elements have a single pull output, so to an element downstream, *Queue*s and schedulers are indistinguishable. We can exploit this property to build *virtual queues*, compound elements that act like queues but implement more complex behavior than FIFO queueing. Figure 10 shows a virtual queue that implements a version of stochastic fairness queueing [McKenney 1990]: packets are hashed into one of several queues that are scheduled round-robin, providing some isolation between competing flows. The IP router can be extended to use stochastic fairness queueing by replacing its *Queue*s with copies of Figure 10.

4.2 Dropping Policies

The *Queue* element implements a simple dropping policy: a configurable maximum length beyond which all incoming packets are dropped. Other policies build on *Queue* rather than replacing it. For example, an independent *RED* element implements random early detection dropping [Floyd and Jacobson 1993]. *RED* contains only drop decision code; this separates the dropping policy from the packet storage policy, allowing either to be independently replaced. It also allows important policy variants through configuration rearrangement.

Random early detection is more likely to drop packets when there is network congestion; a link is considered congested when there are many packets in the queue servicing that link. The *RED* element therefore queries router queue lengths when deciding whether to drop a passing packet. Specifically, it queries the number of packets in the nearest downstream *Storage* elements, which it finds using flow-based router context. (*Storage* is a simple method interface implemented by *Queue* and other elements that store packets.) *RED* can handle one or more downstream *Storage* elements; if there are more than one, it adds their packet counts together to form a single virtual count. This generalization lets it implement variants like RED over multiple queues: a *RED* element placed before the *HashDemux* in Figure 10 would count both of the figure's *Queue*s. A different arrangement implements weighted RED [Cisco Corporation 1999], where packets are dropped with different probabilities depending on their priority; see Figure 11. Finally, *RED* can be positioned after the *Queue*s instead of before them. In this case, it is a pull element and looks upstream

RED HAT 1012
RED HAT V. CAS

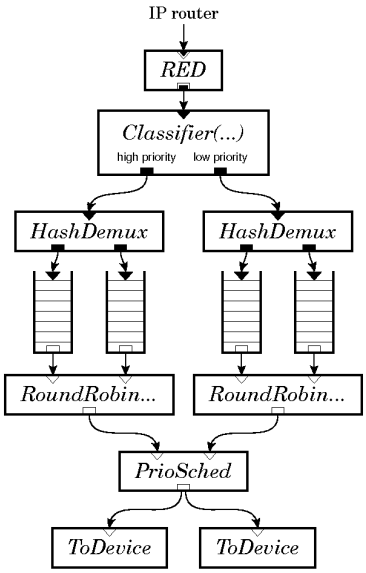


Fig. 12. A complex combination of dropping, queueing, and scheduling. The *Classifier* prioritizes input packets into two virtual queues, each of which implements stochastic fair queueing (Figure 10). *PrioSched* implements priority scheduling on the virtual queues, preferring packets from the left. The router is driving two equivalent T1 lines that pull packets from the same sources, providing a form of load balancing. Finally, *RED*, at the top, implements RED dropping over all four *Queue*s.

rather than downstream for *Storage* elements, creating a strategy like drop-from-front RED [Lakshman et al. 1996].

Simple RED dropping can be added to the IP router by adding a *RED* element before each *Queue*.

4.3 Complex Extensions and Simpler Subsets

Click is equally well suited for building simple and complex packet processors. A network administrator can easily extend the Click IP router to handle specialized routing tasks, or use a subset of its elements for some other application.

For example, imagine a complex IP router with the following requirements:

—two parallel T1 links to a backbone, between which traffic should be load-balanced;

—division of traffic into two priority levels;

—fairness among the connections within each priority level;

—RED dropping driven by the total number of packets queued.

Click's modular scheduling, queueing and dropping policy elements make this easy; Figure 12 shows the extension.

Now consider Figure 13, where a remote mobile host (43.0.0.9) would like to appear as part of its home network (1.0.0.*x*) with the help of a home node on that network (1.0.0.10). A simple Click configuration for this task can reuse several

RED HAT 1012
RED HAT V. CAS

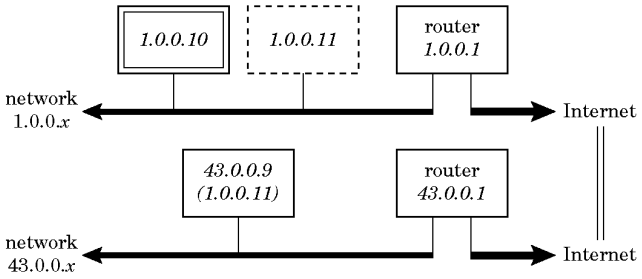


Fig. 13. Network arrangement for a configuration supporting mobility. The mobile host, 43.0.0.9, normally has the IP address 1.0.0.11. Its home node, 1.0.0.10, will encapsulate packets destined for 1.0.0.11 in packets destined for 43.0.0.9.

of the IP router's elements. Figure 14 shows the configuration active at the home node; it proxy-ARPs for the mobile host's home IP address and performs IP-in-IP encapsulation and unencapsulation to communicate with the mobile host.

4.4 Differentiated Services

The Differentiated Services architecture [Blake et al. 1998] provides mechanisms for border and core routers to jointly manage aggregate traffic streams. Diffserv border routers classify and tag packets according to traffic type and ensure that traffic enters the network no faster than allowed; core routers queue and schedule packets based on their tags. The diffserv architecture envisions flexible combinations of classification, tagging, shaping, dropping, queueing, and scheduling functions. We have implemented all of these components as Click elements, which gives router administrators full control over how they are arranged.

For example, Figure 15 implements a realistic diffserv traffic conditioning block. This configuration separates incoming traffic into 4 streams based on the IP header's Differentiated Services Code Point field (DSCP) [Nichols et al. 1998]. The first three streams are rate-limited, while the fourth represents normal best-effort delivery. The rate-limited streams are given priority over the normal stream. From top to bottom in Figure 15, these streams are (A) limited by dropping—the stream is dropped when more than 7,500 packets per second are received on average; (B) shaped—at most 10,000 packets per second are allowed through the *Shaper*, with any excess packets left enqueued; and (C) limited by reclassification—when more than 12,500 packets per second are received, the stream is reclassified as best-effort delivery and sent into the lower priority queue.

4.5 Ethernet Switch

Figure 16 shows a Click configuration unrelated to IP, namely a functional, IEEE 802.1d-compliant Ethernet switch. It acts as a learning bridge and participates with other 802.1d-compliant bridges to determine a spanning tree for the network.

The *EtherSwitch* element can be used alone as a simple, functional learn-

RED HAT 1012
RED HAT V. CAS

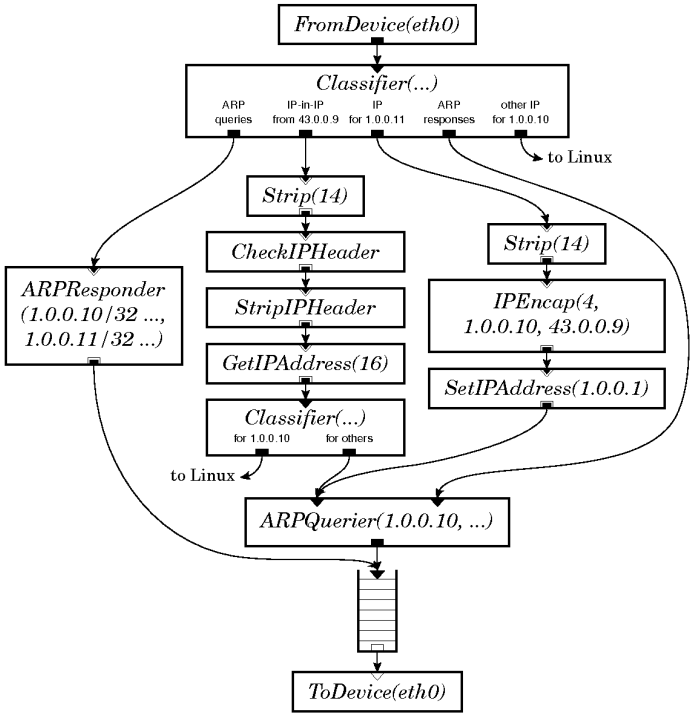


Fig. 14. A configuration for the home node, 1.0.0.10, in the arrangement of Figure 13. The home node proxy-ARPs for the mobile node (*ARPResponder*), unencapsulates packets from the remote node, sending them onto the local network (*CheckIPHeader* path), and performs IP encapsulation for packets destined for the mobile node (*IPEncap* path). Elements not shown ensure that packets generated by 1.0.0.10 itself are properly encapsulated.

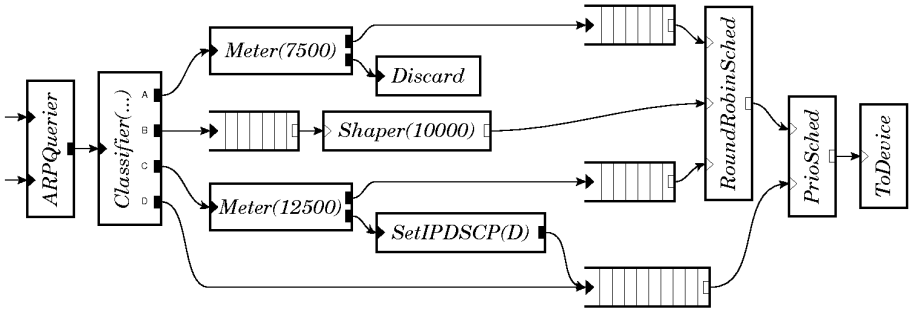


Fig. 15. A sample traffic conditioning block. *Meter*s and *Shaper*s measure traffic rates in packets per second. A, B, C, and D represent DSCP values.

ing bridge. The *EtherSpanTree* and *Suppressor* elements are necessary only to avoid cycles when multiple bridges are used in a LAN. *EtherSpanTree* implements the IEEE 802.1d protocol for constructing a network spanning tree; it works by controlling the *Suppressor* elements. *Suppressor* normally forwards

RED HAT 1012
RED HAT V. CAS

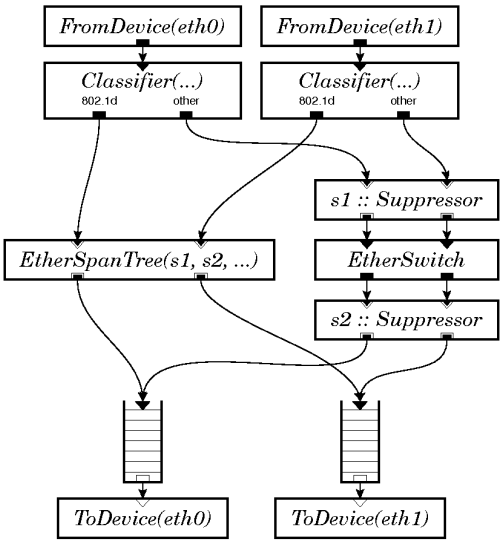


Fig. 16. An Ethernet switch configuration.

packets from each input to the corresponding output, but also exports a method interface for suppressing and unsuppressing individual ports. Packets arriving on a suppressed port are dropped. *EtherSpanTree* uses this interface to prevent *EtherSwitch* from forwarding packets along edges not on the spanning tree. The *Suppressor*s cannot be found using flow-based router context, so the user must specify them by name in *EtherSpanTree*'s configuration string.

5. KERNEL ENVIRONMENT

Click runs as a kernel thread inside a Linux 2.2 kernel. The kernel thread runs the router driver, which loops over the task queue and runs each task. Only interrupts can preempt this thread, but to keep the system responsive, the router driver voluntarily gives up the CPU to Linux from time to time. (The user can configure how often with a *ScheduleLinux* element.) There is generally only one driver thread active at a time. However, during configuration installation, there may be two or more simultaneously active router configurations and driver threads—the new router plus several old ones in the process of dying.

The Click kernel module uses Linux's `/proc` filesystem to communicate with user processes. To bring a router on line, the user creates a configuration description in the Click language and writes it to `/proc/click/config` (or `/proc/click/hotconfig` for a hot-swap installation). Other files in `/proc/click` export information about the currently installed configuration and memory statistics. When installing a router configuration, Click creates a subdirectory under `/proc/click` for each element. This subdirectory contains that element's handlers (Section 2.6). The code that handles accesses to `/proc` runs outside the Click driver thread, in the context of the reading or writing process.

A running Click router contains four important object categories: elements,

RED HAT 1012
RED HAT V. CAS

a router, packets, and timers.

—**Elements.** The system contains an element object for each element in the current configuration, as well as prototype objects for every kind of primitive element that could be used.

—**Router.** The single router object collects information relevant to a given router configuration. It configures the elements, checks that connections are valid, and puts the router on line; it also manages the task queue.

—**Packets.** Click packet data is stored in a single block of memory, as opposed to a BSD `mbuf`-like structure. Packet data is copy-on-write—when copying a packet, the system copies the packet header but not the data. Annotations are stored in the packet header in a fixed static order; there is currently no way to dynamically add a new kind of annotation. In the Linux kernel, Click packet objects are equivalent to `sk_buffs` (Linux's packet abstraction).

—**Timers.** In the Linux kernel, Click timers are implemented with Linux timer queues, which on Intel PCs have .01-second resolution.

5.1 Polling and Device Handling

The original Click system [Morris et al. 1999] shared Linux's interrupt structure and device handling; our goal was to change Linux as little as possible. However, interrupt overhead and device handling dominated that system's performance, consuming about 4.8 and 6.7 μs respectively of the 13 μs required to forward a packet on a 700 MHz Pentium III PC. In addition, Click processed packets at a lower priority than interrupts, leading to receive livelock [Mogul and Ramakrishnan 1997]: with increasing numbers of input packets, interrupt processing eventually starved all other system tasks, leading to reduced throughput.

Click now eliminates interrupts in favor of polling. Device-handling elements —namely, *FromDevice* and *ToDevice*—place themselves on Click's task queue. When activated, *FromDevice* polls its device's receive DMA queue for newly arrived packets; if any are found, it pushes them through the configuration. *ToDevice* examines its device's transmit DMA queue for empty slots, which it fills by pulling packets from its input. These elements also refill the receive DMA list with empty buffers and remove transmitted buffers from the transmit DMA list. The device never interrupts the processor; furthermore, Linux networking code never executes, except as requested by the Click router configuration. Mogul and Ramakrishnan [1997] also used polling to eliminate receive livelock. However, their system left interrupts enabled under light load, while Click is a pure polling system—even infrequent PC interrupts are simply too expensive.

Polling required changes to Linux's structure representing network devices and the drivers for the network devices we used. The new device structure includes, for example, methods to turn interrupts on and off, to poll for received packets, and to clean the transmit DMA ring.

The introduction of polling let us eliminate all programmed I/O (PIO) interaction with the Ethernet controllers. One PIO reenabled receive interrupts after an interrupt was processed, and was not necessary in a polling system.

RED HAT 1012
RED HAT V. CAS

The second PIO was used in the send routine to tell the controller to look for a new packet to transmit. To eliminate it, we configured the network card to periodically check the transmit DMA queue for new packets. As a result, all communication between Click and the controller takes place indirectly through the shared DMA queues in main memory, in the style of the Osiris network controller [Druschel et al. 1994]. This saved roughly two microseconds per packet, as a PIO takes about 1 μs to execute.

These improvements eliminated receive livelock, reduced the total time required to process a packet from 13 μs to 2.8 μs, and increased Click's maximum loss-free forwarding rate of minimum-sized packets more than fourfold. Mogul and Ramakrishnan [1997] found that polling eliminated receive livelock but did not increase peak forwarding rate. We hypothesize that interrupts are relatively much more expensive on PCs than on the Alpha hardware they used.

6. EVALUATION

This section evaluates Click's performance for IP routing and for several extended configurations. We also describe and analyze sources of overhead in Click's design and implementation.

6.1 Experimental Setup

The experimental setup consists of a total of nine Intel PCs running Linux—specifically, a version of Linux 2.2.14 modified to add support for Click's polling device drivers. Of the nine PCs, one is the router host, four are source hosts, and four are destination hosts. The router host has eight 100 Mbit/s Ethernet controllers connected, by point-to-point links, to the source and destination hosts. During a test, each source generates an even flow of UDP packets addressed to a corresponding destination; the router is expected to get them there. The tests use version 1.1 of the Click software.

The router host has a 700 MHz Intel Pentium III CPU and an Intel L440GX+ motherboard. Its Ethernet controllers are DEC 21140 Tulip 100 Mbit/s PCI controllers [Digital Equipment Corporation 1998] on multi-port cards, split across the motherboard's two independent PCI buses. The Pentium III has a 16 KB L1 instruction cache, a 16 KB L1 data cache, and a 256 KB L2 unified cache. The source and destination hosts have 733 MHz Pentium III CPUs and 200 MHz Pentium Pro CPUs, respectively. Each has one DEC 21140 Ethernet controller. The source-to-router and router-to-destination links are point-to-point full-duplex 100 Mbit/s Ethernet.

The source hosts generate UDP packets at specified rates, and can generate up to 147,900 64-byte packets per second. The destination hosts count and discard the forwarded UDP packets. Each 64-byte UDP packet includes Ethernet, IP, and UDP headers as well as 14 bytes of data and the 4-byte Ethernet CRC. When the 64-bit preamble and 96-bit inter-frame gap are added, a 100 Mbit/s Ethernet link can carry up to 148,800 such packets per second.

6.2 Forwarding Rates

We characterize performance by measuring the rate at which a router can forward 64-byte packets over a range of input rates. Minimum-size packets stress

RED HAT 1012
RED HAT V. CAS

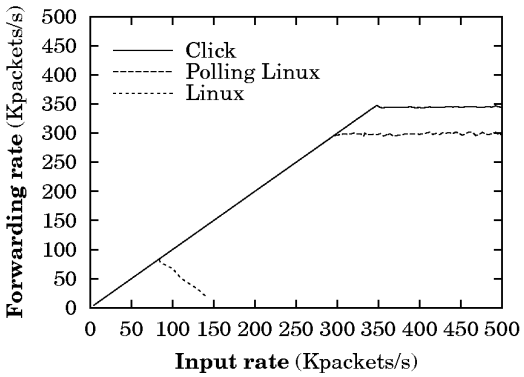


Fig. 17. Forwarding rate as a function of input rate for Click and Linux IP routers (64-byte packets).

the router harder than larger packets; the CPU and several other bottleneck resources are consumed in proportion to the number of packets forwarded, not in proportion to bandwidth. Plotting forwarding rate versus input rate indicates both the maximum loss-free forwarding rate (MLFFR) and the router's behavior under overload. An ideal router would emit every input packet regardless of input rate, corresponding to the line $y = x$.

Figure 17 compares Click's and Linux's performance for IP routing. The line marked "Click" shows the performance of an eight-interface version of the Click IP router configuration in Figure 8. This configuration has a total of 161 elements (19 elements per interface plus 9 elements shared by all interfaces). The "Linux" and "Polling Linux" lines show the performance of Linux IP routing; "Linux" uses the standard interrupting device drivers, while "Polling Linux" was modified to use Click's polling drivers.

Click's maximum loss-free forwarding rate is 333,000 packets per second; its peak forwarding rate, attainable with bursty senders, is 360,000 packets per second. When these rates are exceeded, the Ethernet controllers report that they are discarding received packets due to insufficient receive DMA descriptors, which suggests that the bottleneck is the router's CPU, not its PCI bus bandwidth.

Standard Linux's MLFFR is 84,000 packets per second; its output rate declines as the input rate increases due to receive livelock. Polling Linux's MLFFR of 284,000 packets per second is over three times standard Linux's MLFFR, which demonstrates the advantages of avoiding interrupts and handling device I/O efficiently. Polling Linux is slightly slower than Click, perhaps because Linux's routing table lookup algorithms are slower, but more scalable.

An otherwise idle Click IP router forwards 64-byte packets with a one-way latency of 29 μs. This number was calculated by taking the round-trip ping time through the router (measured with `tcpdump`), subtracting the round-trip ping time with the router replaced by a wire, and dividing by two. 5.8 μs of the 29 are due to the time required to transmit a 64-byte Ethernet packet at 100 megabits per second. 10 μs are due to the costs of polling eight Ethernet controllers, all

RED HAT 1012
RED HAT V. CAS

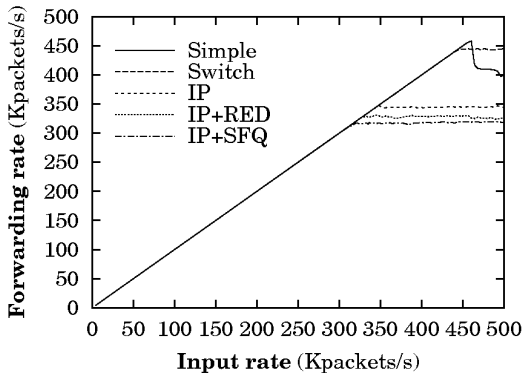


Fig. 18. Forwarding rate as a function of input rate for extended IP router configurations and some non-IP routers (64-byte packets).

but two of which are idle in this test. 2.5 μs of delay is the expected amount of time the Tulip controller will take to poll for a new packet to transmit. The latency of a router running standard Linux IP code is 33 μs.

Figure 18 shows the performance of some extended IP router configurations and some non-IP-router configurations. The "Simple" line shows the performance of the simplest possible Click router, which forwards input packets directly from input to output with no intervening processing. The MLFFR for this configuration is 452,000 packets per second. It is limited by the PCI bus, not the CPU, which explains its different behavior under overload. The "Switch" line corresponds to the Ethernet switch configuration of Figure 16; its MLFFR is close to that of "Simple". The "IP" line, repeated from Figure 17, shows the performance of the base IP router configuration. The "IP+RED" line corresponds to an IP router in which a *RED* element is inserted before each *Queue*. No packets were dropped by RED in the test, since the router's output links are as fast as its inputs. The "IP+SFQ" line shows the performance of an IP router with each *Queue* replaced by a stochastic fair queue—namely, a version of Figure 10 with four component *Queue*s. As router complexity increases from simple forwarding through IP routing to IP routing with stochastic fair queueing, Click's performance drops only gradually.

This test configuration shows both Click and Linux in a better light than might be seen in a real network. The tests did not involve fragmentation, IP options, or ICMP errors, though Figure 8 has all the code needed to handle these. Increasing the number of hosts involved might slow Click down by increasing the number of ARP table entries. Increasing the routing table size would also decrease performance, a problem addressed by previous work on fast lookup in large tables [Degermark et al. 1997; Waldvogel et al. 1997]. The configuration is nevertheless sufficient to analyze the architectural aspects of Click's performance.

RED HAT 1012
RED HAT V. CAS

Table I. Microbenchmarks of Tasks Involved in the Click IP Forwarding Path

Task	**Time** (ns/packet)
Polling packet	528
Refill receive DMA ring	90
Push through Click forwarding path	1565
Pull from Click queue	103
Enqueue packet for transmit	161
Clean transmit DMA ring	351
Total	**2798**

6.3 Analysis

Table I breaks down the cost of forwarding a packet through a Click router. Costs were measured in nanoseconds by Pentium III cycle counters [Intel Corporation 1996]; each cost is the accumulated cost for all packets in a 10-second run divided by the number of packets forwarded. "Polling packet" is the time *FromDevice* spends taking a packet from the Tulip's receive DMA ring. "Refill receive DMA ring" is the time *FromDevice* spends replacing the DMA descriptor of the packet it just received with a new descriptor, so that the Tulip may receive a new packet. The Click IP forwarding path (see Figure 8) is divided into a push path and a pull path. The push path involves 15 elements; it begins when an input packet is emitted by some *FromDevice* element and ends when the packet reaches the *Queue* element before the appropriate transmitting device. The pull path involves just two elements—when a *ToDevice* element is ready to send, it pulls a packet from a *Queue*. The time consumed by Linux's equivalent of the push path is 1.65 μs, slightly more than Click's 1.57 μs. "Enqueue packet for transmit" is the time *ToDevice* spends enqueueing a packet onto the Tulip's transmit DMA ring. "Clean transmit DMA ring" is the time *ToDevice* spends removing DMA descriptors of transmitted packets. Overall, Click code takes 60% of the time required to process a packet; device code takes the other 40%.

Table II breaks down the 1.57 μs push cost by element. Each element's cost is the difference between the Pentium III cycle counter value before and after executing the element, decreased by the time required to read the cycle counter; it includes the virtual function call(s) that move a packet from one element to the next. (*IPFragmenter* is so inexpensive because it does nothing—no packets need to be fragmented in this test—and its packet handoff requires one virtual function call. Other elements that do nothing in this test, such as *IPGWOptions*, have agnostic ports that incur two virtual function calls per packet handoff.) The *FromDevice* and *Queue* elements are not included.

The total cost of 2798 ns measured with performance counters implies a maximum forwarding rate of about 357,000 packets per second, consistent with the observed peak forwarding rate of 360,000 packets per second.

Forwarding a packet through Click incurs five L2 data cache misses (measured using Pentium III performance counters): one to read the receive DMA descriptor, two to read the packet's Ethernet and IP headers, and two to remove the packet from the transmit DMA queue and put it into a pool of reusable packets. Each of these loads takes about 112 nanoseconds. Click runs without

RED HAT 1012
RED HAT V. CAS

Table II. Execution Times of Elements Involved in IP Forwarding, in Nanoseconds per Packet

Element	**Time** (ns)
Classifier	70
Paint	77
Strip	67
CheckIPHeader	457
GetIPAddress	120
LookupIPRoute	140
DropBroadcasts	77
CheckPaint	67
IPGWOptions	63
FixIPSrc	63
DecIPTTL	119
IPFragmenter	29
ARPQuerier	106
Subtotal	**1455**

incurring any other L2 data or instruction cache misses. 1295 instructions are retired during the forwarding of a packet, implying that significantly more complex Click configurations could be supported without exhausting the Pentium III's 16K L1 instruction cache.

6.4 Overhead of Modularity

Click's modularity imposes performance costs in two ways: the overhead of passing packets between elements, and the overhead of unnecessarily general element code.

Passing a packet from one element to the next causes either one or two virtual function calls, where a virtual function call includes loading the relevant function pointer from a virtual function table as well as an indirect jump through that function pointer. Indirect jumps take about 10 nanoseconds if the Pentium III's branch target buffer successfully predicts the target address, but can take dozens of nanoseconds if it fails. Most elements in the IP router use the simplified agnostic-port interface that causes two virtual function calls per packet handoff; the CPU usually predicts one call but doesn't predict the other. As a result, the total cost of packet handoff is about 70 nanoseconds. Thus, adding an empty element to a configuration graph usually adds 70 nanoseconds of overhead, and the IP router, with 16 elements on its forwarding path, has about 1 μs of overhead. This overhead is avoidable—we have built tools that can automatically eliminate all virtual function calls from a Click configuration [Kohler 2000].

Several elements in the IP router configuration were implemented with more generality than the IP router requires. The best example is *Classifier*, which the IP router uses to classify Ethernet packets into ARP queries, ARP responses, and IP packets. However, *Classifier* can handle many classification tasks; at run time, it interprets a small data structure (derived from its configuration string) that tells it which packet data to examine. An element specialized for the IP router's classification task could use straight-line code with compiled-in

RED HAT 1012
RED HAT V. CAS

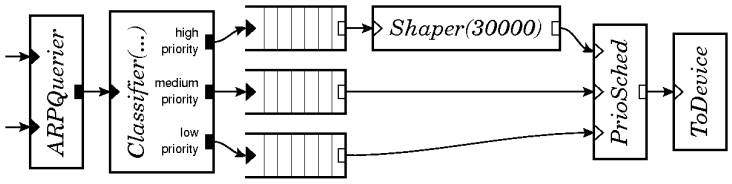


Fig. 19. Another Click diffserv traffic conditioning block. This configuration fragment implements strict priority with three priority levels; in addition, the high-priority stream is rate-limited to 30,000 packets per second.

constants, avoiding interpretation overhead. We measured this overhead by writing such a specialized classifier. This element cost 24% less CPU time per packet than the generic *Classifier*, but even the generic *Classifier* takes only 4% of the per-packet CPU time expended inside the Click configuration. Since the rest of the IP router's elements offer less opportunity for specialization, we conclude that element generality has a relatively small effect on Click's performance.

To analyze the combined effects of these overheads, we wrote two single elements that combine much of the processing in Figure 8. These elements reduce the number of virtual function calls on the forwarding path, specialize some of the more general elements, and offer the compiler better opportunities for optimization. One element implements the functionality of *Paint*, *Strip*, *CheckIPHeader*, and *GetIPAddress*; the other, the functionality of *DropBroadcasts*, *CheckPaint*, *IPGWOptions*, *FixIPSrc*, *DecIPTTL*, and *IPFragmenter*. The resulting configuration is equivalent to Figure 8, but has only eight elements on the forwarding path instead of sixteen. The new configuration's push path processes an IP packet in 1.03 μs instead of 1.57 μs. When we add eight distinct do-nothing elements to the forwarding path of this new configuration, restoring the number of elements to sixteen, the packet processing time rises to 1.58 μs. This suggests that the entire reduction from 1.57 μs to 1.03 μs is due to reducing the number of virtual function calls.

6.5 Differentiated Services Evaluation

Section 4.4 showed that Click can conveniently model Differentiated Services configurations; this section shows that Click can enforce diffserv policies using only packet scheduling elements like *PrioSched*. The test configuration involves three source hosts sending packets marked as high, medium, and low priority respectively. The router has a single 100 Mbit/s Ethernet output link. It uses the configuration in Figure 8 with each *Queue* replaced with Figure 19. The configuration queues each priority level separately; it also rate-limits the highest priority traffic to no more than 30,000 packets per second. The *PrioSched* packet scheduling element implicitly schedules CPU time as well as link bandwidth.

Figure 20 shows the results. The per-source sending rate varies along the x axis; the total input load on the router is three times the x-axis value. The y axis indicates how many packets per second the router forwarded from each of the sources. Once the per-source input rate rises above 30,000 packets per

RED HAT 1012
RED HAT V. CAS

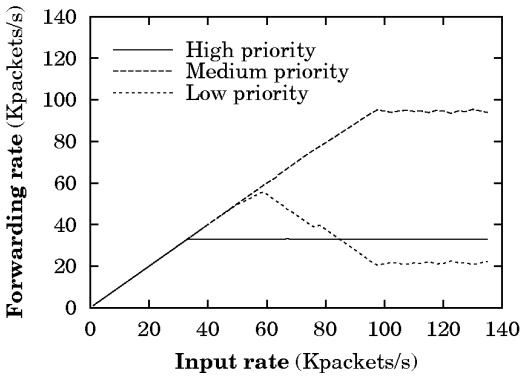


Fig. 20. Performance of the Click diffserv configuration in Figure 19. Three traffic sources send 64-byte UDP packets with three different priorities into a router with a single 100 Mbit/s output link. The sources send equal numbers of high, medium, and low-priority packets; the x axis indicates the input rate for each priority level.

second, the router starts dropping high-priority packets, since they are rate-limited. Above 57,000 packets per second, the output link is saturated, and medium priority packets start taking bandwidth away from low. Above 100,000 packets per second, the router runs out of CPU time, forcing the Ethernet controllers to drop packets. Since the controllers don't understand priority, they drop medium-priority packets as well as low. Each medium-priority packet that a controller drops leaves room on the output link for a low-priority packet, which is why the low-priority forwarding rate isn't driven down to zero.

There is probably no way to achieve perfect packet scheduling under overload; instead, a router should be engineered with sufficient processing power to carry all packets to the point where the priority decision is made. The Click user can approximate such a design by moving the classification and per-priority queues of Figure 19 earlier in the configuration—that is, nearer to *FromDevice* than *ToDevice*. This wastes less CPU time on lower priority packets that will eventually be dropped, because they travel through less of the configuration before reaching the drop point. Even with the priority decision late in the configuration, Click enforces priority reasonably well under overload and very well with ordinary loads.

7. RELATED WORK

The x-kernel [Hutchinson and Peterson 1991] is a framework for implementing and composing network protocols. It is intended for use at end nodes, where packet motion is vertical (between the network and user level) rather than horizontal (between network interfaces). Like a Click router, an x-kernel configuration is a graph of processing nodes, and packets are passed between nodes by virtual function calls. However, an x-kernel configuration graph is always acyclic and layered, as x-kernel nodes were intended to represent protocols in a protocol stack. This prevents cyclic configurations like our IP router. The x-kernel's inter-node communication protocols are more complex than Click's.

RED HAT 1012
RED HAT V. CAS

Connections between nodes are bidirectional—packets travel up the graph to user level and down the graph to the network. Packets pass alternately through "protocol" nodes and "session" nodes, where the session nodes correspond to end-to-end network connections like TCP sessions. Bidirectional connections and session nodes are irrelevant to most routers.

Scout [Mosberger and Peterson 1996; Peterson et al. 1999] was designed for use as an operating system at end nodes, but it is better suited for routing than the *x*-kernel; for example, cyclic configurations are partially supported, and session nodes are not mandatory. Execution in Scout is centered on "paths", sequences of packet processing functions. Each path has implicit queues on its inputs and outputs; it is not clear, therefore, how many queues would appear in a complex configuration like the IP router, which is not amenable to linearization. Each path is run by a single thread and has a CPU priority, and packets are classified into the correct path as early as possible. For example, TCP/IP packets containing MPEG data can be routed immediately to the correct path, whose scheduling behavior might be specialized for MPEG. If desired, Click primitives can support some uses of paths: an early *Classifier* element could send MPEG-in-TCP-in-IP-in-Ethernet traffic into a special *Queue*, and the element that took packets off that *Queue* could be scheduled appropriately for MPEG data. However, Scout supports per-flow paths, where a path object is created for each individual network flow. This allows its CPU scheduler to treat flows independently; Click's CPU scheduler can only treat flow classes independently.

The UNIX System V STREAMS system [Ritchie 1984] is also built from composable packet processing modules. STREAMS modules include implicit queueing by default. Each module must be prepared for the next module's queue to fill up, and to respond by queueing or discarding or deferring the processing of incoming packets. Modules with multiple inputs or outputs must also make packet scheduling decisions. STREAMS' tendency to spread scheduling and queueing logic throughout the configuration conflicts with a router's need for precise control over these functions.

FreeBSD contains a modular networking system called Netgraph [Elischer and Cobbs 1998] whose nodes strongly resemble Click elements. Netgraph was designed for composing coarse-grained modules such as PPP encapsulation and other framing protocols. Configurations are built up dynamically; there are commands that add and delete nodes, and that connect and disconnect "hooks" (the equivalent of ports), but no external specification for a configuration. This supports on-line configuration modification somewhat more naturally than Click, but there is no way to analyze a Netgraph configuration off line or install a new configuration atomically.

ALTQ [Cho 1998], a system for configurable traffic management, is also shipped with FreeBSD. It contains several sophisticated queueing policies, but its configurability is limited to the specification of one queueing policy per output interface.

The router plugins system [Decasper et al. 1998; Decasper 1999] was designed to make IP routers more extensible. A router plugin is a software module executed when a classifier matches a particular flow. These classifiers can

RED HAT 1012
RED HAT V. CAS

be installed at any of several "gates". However, gates are fixed points in the IP forwarding path; there is no way to add new gates or to control the path itself.

To the best of our knowledge, commercial routers are difficult to extend, either because they use specialized hardware [Newman et al. 1998; Partridge et al. 1998] or because their software is proprietary. However, open router software does not necessarily make a router easily extensible. A network administrator could, in principle, implement new routing functions in Linux, but we expect few administrators have the time or ability to modify the monolithic Linux kernel. Click's modularity makes it easy to take advantage of its extensibility.

The active networking research program allows anyone to write code that will affect a router [Tennenhouse et al. 1997; Smith et al. 1999]. Click allows a trusted user to change any aspect of a router; active networking allows untrusted packets to decide how they should be routed. The two approaches are complementary, and Click may provide a good platform for active network research.

Click's polling implementation and device driver improvements were derived from previous work [Druschel et al. 1994; Mills 1988; Mogul and Ramakrishnan 1997; Wroclawski 1997].

8. CONCLUSION

Click is an open, extensible, and configurable router framework. The Click IP router demonstrates that real routers can be built from small, modular elements, and our performance analysis shows that modularity is compatible with good forwarding performance for PC hardware. IP router extensions for scheduling and dropping policies, complex queueing, and Differentiated Services simply require adding and rearranging a couple elements. Finally, Click is flexible enough to support other packet processing applications. The Click system is free software; it is available for download at `http://www.pdos.lcs.mit.edu/click/`.

APPENDIX

ELEMENT GLOSSARY

This section lists 38 of the approximately 130 elements included with version 1.1 of the Click software. Each entry follows this format:

ElementName(configuration arguments) Push, pull, or agnostic (specifies port types). Port descriptions (packet types and numbers of ports). Description.

More detailed descriptions are accessible on line at `http://www.pdos.lcs.mit.edu/click/doc/`.

ARPQuerier(...) Push. First input takes IP packets, second input takes ARP responses with Ethernet headers. Output emits ARP queries and IP-in-Ethernet packets. Uses ARP to find the Ethernet address corresponding to each input IP packet's destination IP address annotation; encapsulates the packet in an Ethernet header with that destination Ethernet address.

RED HAT 1012
RED HAT V. CAS

ARPResponder(ip eth, ...) Agnostic. Input takes ARP queries, output emits ARP responses. Responds to ARP queries for IP address *ip* with the static Ethernet address *eth*.

CheckIPHeader(...) Agnostic. Input takes IP packets. Discards packets with invalid IP length, source address, or checksum fields; forwards valid packets unchanged.

CheckPaint(p) Agnostic. Input takes any packet. Forwards packets with paint annotation *p* to both outputs; otherwise just to first.

Classifier(...) Push. Input takes any packet. Examines packet data according to a set of classifiers, one classifier per output port. Forwards packet to output port corresponding to the first classifier that matched. Example classifier: "*12/0800*" checks that the packet's data bytes 12 and 13 contain values 8 and 0, respectively. See also *IPClassifier*.

DecIPTTL Agnostic. Input takes IP packets. Decrements input packets' IP time-to-live field. If the packet is still live, incrementally updates IP checksum and sends modified packet to first output; if it has expired, sends unmodified packet to second output.

Discard Push. Discards all input packets.

DropBroadcasts Agnostic. Input takes any packet. Discards packets that arrived as link-level broadcasts; forwards others unchanged.

EtherSpanTree Agnostic. Inputs take IEEE 802.1d messages. Implements the IEEE 802.1d spanning tree algorithm for Ethernet switches.

EtherSwitch Push. Inputs take Ethernet packets; one input per output. Learning, forwarding Ethernet switch. Learns the interfaces corresponding to Ethernet addresses by examining input packets' source addresses; forwards packets to the correct output port, or broadcasts them if the destination Ethernet address is not yet known.

FixIPSrc(ip) Agnostic. Input takes IP packets. Sets the IP header's source address field to the static IP address *ip* if the packet's Fix IP Source annotation is set; forwards other packets unchanged.

FromDevice(devicename) Push. No inputs. Sends packets to its single output as they arrive from a Linux device driver.

FromLinux(devicename, ip/netmask) Push. No inputs. Installs into Linux a fake Ethernet device *devicename* and a routing table entry that sends packets for *ip*/*netmask* to that fake device. The result is that packets generated at the router host and destined for *ip*/*netmask* are emitted on *FromLinux*'s single output as they arrive from Linux.

GetIPAddress(16) Agnostic. Input takes IP packets. Copies the IP header's destination address field (offset 16 in the IP header) into the destination IP address annotation; forwards packets unchanged.

HashDemux(offset, length) Push. Input takes any packet; arbitrary number of outputs. Forwards packet to one of its outputs, chosen by a hash bytes [*offset*, *offset* + *length*) of the packet's data.

ICMPError(ip, type, code) Agnostic. Input takes IP packets, output emits ICMP error packets. Encapsulates first part of input packet in ICMP error

RED HAT 1012
RED HAT V. CAS

header with source address *ip*, error type *type*, and error code *code*. Sets the Fix IP Source annotation on output packets.

IPClassifier(...) Push. Input takes IP packets. Examines packet data according to a set of classifiers, one classifier per output port. Forwards packet to output port corresponding to the first classifier that matched. Example classifier: *"ip src 1.0.0.1 and dst tcp port www"* checks that the packet's source IP address is 1.0.0.1, its IP protocol is 6 (TCP), and its destination TCP port is 80. See also *Classifier*.

IPEncap(p, ipsrc, ipdst) Agnostic. Input takes any packet, output emits IP packets. Encapsulates input packets in an IP header with protocol p, source address *ipsrc*, and destination address *ipdst*.

IPFragmenter(mtu) Push. Input takes IP packets. Fragments IP packets larger than *mtu*; sends fragments, and packets smaller than *mtu*, to first output. Too-large packets with the don't-fragment IP header flag set are sent unmodified to the second output.

IPGWOptions Agnostic. Input takes IP packets. Processes IP Record Route and Timestamp options; packets with invalid options are sent to the second output.

LookupIPRoute(table) Push. Input takes IP packets with valid destination IP address annotations. Arbitrary number of outputs. Looks up input packets' destination annotations in a static routing table specified in the configuration string. Forwards each packet to the output port specified in the resulting routing table entry; sets its destination annotation to the resulting gateway address, if any.

Meter(r) Push. Input takes any packet. Sends packets to first output if recent input rate averages $< r$, second output otherwise. Multiple rate arguments are allowed.

Paint(p) Agnostic. Input takes any packet. Sets each input packet's paint annotation to p before forwarding it.

PrioSched Pull. Inputs take any packet; one output port, arbitrary number of input ports. Responds to pull requests by trying inputs in numeric order, returning the first packet it receives. Thus, lower-numbered inputs have priority.

Queue(n) Push input, pull output. Input takes any packet. Stores packets in a FIFO queue; maximum queue capacity is n.

RED(min-thresh, max-thresh, max-p) Agnostic. Input takes any packet. Drops packets probabilistically according to the Random Early Detection algorithm [Floyd and Jacobson 1993] with the given parameters; forwards other packets unchanged. Examines nearby router queue lengths when making its drop decision.

RoundRobinSched Pull. Inputs take any packet; one output port, arbitrary number of input ports. Responds to pull requests by trying inputs in round-robin order; returns the first packet it receives. It first tries the input port just after the one that succeeded on the last pull request.

RED HAT 1012
RED HAT V. CAS

ScheduleInfo(elementname param, ...) No inputs or outputs. Specifies elements' initial task queue scheduling parameters by element name. Each scheduling parameter is a real number; an element with parameter $2p$ will be scheduled twice as often as an element with parameter p.

ScheduleLinux No inputs or outputs. Places itself on the task queue, and returns to Linux's scheduler every time it is run.

SetIPAddress(ip) Agnostic. Input takes IP packets. Sets each packet's destination IP address annotation to the static IP address *ip*. Forwards modified packets to first output.

SetIPDSCP(c) Agnostic. Input takes IP packets. Sets each packet's IP diffserv code point field [Nichols et al. 1998] to c and incrementally updates the IP header checksum. Forwards modified packets to first output.

Shaper(n) Pull. Input takes any packet. Simple pull traffic shaper: forwards at most n pull requests per second to its input. Pull requests over that rate get a null pointer in response.

Strip(n) Agnostic. Input takes any packet. Strips off each packet's first n bytes; forwards modified packets to first output.

StripIPHeader Agnostic. Input takes IP packets. Strips off each packet's IP header, including any options; forwards modified packets to first output.

Suppressor Agnostic. Inputs take any packet; arbitrary number of input ports, same number of output ports. Normally forwards packets arriving on each input port to the corresponding output port. A method interface allows other elements to ask *Suppressor* to drop packets arriving on particular input ports.

Tee(n) Push. Input takes any packet; n output ports. Forwards each input packet to all n output ports.

ToDevice(device) Pull. Input takes Ethernet packets; no outputs. Hands packets to a Linux device driver for transmission. Activates pull requests only when the device is ready.

ToLinux Push. Input takes Ethernet packets; Linux will ignore the Ethernet header except for the protocol field. No outputs. Hands input packets to Linux's default network input software.

ACKNOWLEDGMENTS

Massimiliano Poletto helped us enormously with tests and benchmarks. We also thank Alex Snoeren for his work on the IPsec elements, Chuck Blake for help with hardware, and Hari Balakrishnan, Daniel Jackson, Greg Minshall, John Wroclawski, Chandu Thekkath, and the anonymous reviewers for their helpful comments.

REFERENCES

BAKER, F. 1995. Requirements for IP Version 4 routers. RFC 1812 (June), Internet Engineering Task Force. `ftp://ftp.ietf.org/rfc/rfc1812.txt`.

BLAKE, S., BLACK, D., CARLSON, M., DAVIES, E., WANG, Z., AND WEISS, W. 1998. An architecture for differentiated services. RFC 2475 (Dec.), Internet Engineering Task Force. `ftp://ftp.ietf.org/rfc/rfc2475.txt`.

RED HAT 1012
RED HAT V. CAS

CHO, K. 1998. A framework for alternate queueing: towards traffic management by PC-UNIX based routers. In *Proc. USENIX 1998 Annual Technical Conference* (June 1998), pp. 247–258.

CISCO CORPORATION. 1999. Distributed WRED. Technical report. `http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/wred.htm`, as of January 2000.

CLARK, D. 1985. The structuring of systems using upcalls. In *Proc. of the 10th ACM Symposium on Operating Systems Principles (SOSP)* (Dec. 1985), pp. 171–180.

DECASPER, D., DITTIA, Z., PARULKAR, G., AND PLATTNER, B. 1998. Router plugins: A software architecture for next generation routers. In *Proc. ACM SIGCOMM Conference (SIGCOMM '98)* (Oct. 1998), pp. 229–240.

DECASPER, D. S. 1999. *A software architecture for next generation routers*. Ph. D. thesis, Swiss Federal Institute of Technology, Zurich.

DEGERMARK, M., BRODNIK, A., CARLSSON, S., AND PINK, S. 1997. Small forwarding tables for fast routing lookups. In *Proc. ACM SIGCOMM Conference (SIGCOMM '97)* (Oct. 1997), pp. 3–14.

DIGITAL EQUIPMENT CORPORATION. 1998. DIGITAL Semiconductor 21140A PCI Fast Ethernet LAN Controller Hardware Reference Manual. `http://developer.intel.com/design/network/manuals`.

DRUSCHEL, P., PETERSON, L., AND DAVIE, B. 1994. Experiences with a high-speed network adaptor: A software perspective. In *Proc. ACM SIGCOMM Conference (SIGCOMM '94)* (Aug. 1994), pp. 2–13.

ELISCHER, J. AND COBBS, A. 1998. The Netgraph networking system. Technical report (Jan.), Whistle Communications. `http://www.elischer.com/netgraph/`, as of July 2000.

FLOYD, S. AND JACOBSON, V. 1993. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Networking 1*, 4 (Aug.), 397–413.

HUTCHINSON, N. C. AND PETERSON, L. L. 1991. The *x*-kernel: an architecture for implementing network protocols. *IEEE Trans. Software Engineering 17*, 1 (Jan.), 64–76.

INTEL CORPORATION. 1996. Pentium Pro Family Developer's Manual, Volume 3. `http://developer.intel.com/design/pro/manuals`.

KOHLER, E. 2000. *The Click modular router*. Ph. D. thesis, Laboratory for Computer Science, Massachusetts Institute of Technology.

LAKSHMAN, T. V., NEIDHARDT, A., AND OTT, T. J. 1996. The drop from front strategy in TCP and in TCP over ATM. In *Proc. IEEE Infocom*, Volume 3 (March 1996), pp. 1242–1250.

MCCANNE, S. AND JACOBSON, V. 1993. The BSD packet filter: A new architecture for user-level packet capture. In *Proc. Winter 1993 USENIX Conference* (Jan. 1993), pp. 259–269.

MCKENNEY, P. E. 1990. Stochastic fairness queueing. In *Proc. IEEE Infocom*, Volume 2 (June 1990), pp. 733–740.

MILLS, D. L. 1988. The Fuzzball. In *Proc. ACM SIGCOMM Conference (SIGCOMM '88)* (Aug. 1988), pp. 115–122.

MOGUL, J. C. AND RAMAKRISHNAN, K. K. 1997. Eliminating receive livelock in an interrupt-driven kernel. *ACM Trans. Computer Systems 15*, 3 (Aug.), 217–252.

MORRIS, R., KOHLER, E., JANNOTTI, J., AND KAASHOEK, M. F. 1999. The Click modular router. In *Proc. of the 17th ACM Symposium on Operating Systems Principles (SOSP)* (Dec. 1999), pp. 217–231.

MOSBERGER, D. AND PETERSON, L. L. 1996. Making paths explicit in the Scout operating system. In *Proc. 2nd Symposium on Operating Systems Design and Implementation (OSDI '96)* (Oct. 1996), pp. 153–167.

NEWMAN, P., MINSHALL, G., AND LYON, T. L. 1998. IP switching—ATM under IP. *IEEE/ACM Trans. Networking 6*, 2 (April), 117–129.

NICHOLS, K., BLAKE, S., BAKER, F., AND BLACK, D. 1998. Definition of the Differentiated Services field (DS field) in the IPv4 and IPv6 headers. RFC 2474 (Dec.), Internet Engineering Task Force. `ftp://ftp.ietf.org/rfc/rfc2474.txt`.

PARTRIDGE, C., CARVEY, P. P., BURGESS, E., CASTINEYRA, I., CLARKE, T., GRAHAM, L., HATHAWAY, M., HERMAN, P., KING, A., KOHALMI, S., MA, T., MCALLEN, J., MENDEZ, T., MILLIKEN, W. C., PETTYJOHN, R., ROKOSZ, J., SEEGER, J., SOLLINS, M., STORCH, S., TOBER, B., TROXEL, G. D., WAITZMAN, D.,

RED HAT 1012
RED HAT V. CAS

AND WINTERBLE, S. 1998. A 50-Gb/s IP router. *IEEE/ACM Trans. Networking 6*, 3 (June), 237–248.

PETERSON, L. L., KARLIN, S. C., AND LI, K. 1999. OS support for general-purpose routers. In *Proc. 7th Workshop on Hot Topics in Operating Systems (HotOS-VII)* (March 1999), pp. 38–43. IEEE Computer Society Technical Committee on Operating Systems.

POSTEL, J. 1981a. Internet Protocol. RFC 791 (Sept.), Internet Engineering Task Force. `ftp://ftp.ietf.org/rfc/rfc0791.txt`.

POSTEL, J. 1981b. Internet Control Message Protocol. RFC 792 (Sept.), Internet Engineering Task Force. `ftp://ftp.ietf.org/rfc/rfc0792.txt`.

RITCHIE, D. M. 1984. A stream input-output system. *AT&T Bell Laboratories Technical Journal 63*, 8 (Oct.), 1897–1910.

SMITH, J. M., CALVERT, K. L., MURPHY, S. L., ORMAN, H. K., AND PETERSON, L. L. 1999. Activating networks: a progress report. *IEEE Computer 32*, 4 (April), 32–41.

TENNENHOUSE, D. L., SMITH, J. M., SINCOSKIE, W., WETHERALL, D. J., AND MINDEN, G. J. 1997. A survey of active network research. *IEEE Communications Magazine 35*, 1 (Jan.), 80–86.

WALDSPURGER, C. A. AND WEIHL, W. E. 1995. Stride scheduling: deterministic proportional-share resource management. Technical Memo MIT/LCS/TM-528 (June), MIT Laboratory for Computer Science.

WALDVOGEL, M., VARGHESE, G., TURNER, J., AND PLATTNER, B. 1997. Scalable high speed IP routing lookups. In *Proc. ACM SIGCOMM Conference (SIGCOMM '97)* (Oct. 1997), pp. 25–38.

WROCLAWSKI, J. 1997. Fast PC routers. Technical report (Jan.), MIT LCS Advanced Network Architecture Group. `http://mercury.lcs.mit.edu/PC-Routers/pcrouter.html`, as of July 2000.

Received March 2000; revised July 2000; accepted July 2000

RED HAT 1012
RED HAT V. CAS

Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks

Sung-Ju Lee
Internet & Mobile Systems Labs
Hewlett-Packard Laboratories
Palo Alto, CA 94304-1126
sjlee@hpl.hp.com

Mario Gerla
Computer Science Department
University of California
Los Angeles, CA 90095-1596
gerla@cs.ucla.edu

Abstract *– In recent years, routing has been the most focused area in ad hoc networks research. On-demand routing in particular, is widely developed in bandwidth constrained mobile wireless ad hoc networks because of its effectiveness and efficiency. Most proposed on-demand routing protocols however, build and rely on single route for each data session. Whenever there is a link disconnection on the active route, the routing protocol must perform a route recovery process. In QoS routing for wired networks, multiple path routing is popularly used. Multiple routes are however, constructed using link-state or distance vector algorithms which are not well-suited for ad hoc networks. We propose an on-demand routing scheme called Split Multipath Routing (SMR) that establishes and utilizes multiple routes of maximally disjoint paths. Providing multiple routes helps minimizing route recovery process and control message overhead. Our protocol uses a per-packet allocation scheme to distribute data packets into multiple paths of active sessions. This traffic distribution efficiently utilizes available network resources and prevents nodes of the route from being congested in heavily loaded traffic situations. We evaluate the performance of our scheme using extensive simulation.*

I. INTRODUCTION

An ad hoc network is an instantly deployable wireless network without any base station or infrastructure support. Because ad hoc networks can be easily deployed, they are developed in applications such as automated battlefields, search and rescue, crowd control, and disaster recovery. Network hosts in ad hoc networks are equipped with packet radios for communications between one another. Because these packet radios have a short propagation range (150 - 250 meters in an open field), the route is multihop when the communication pair is not within each other's proximity. Nodes are operating in power limited batteries, and bandwidth is constrained in wireless networks. Moreover, all nodes can be mobile, and the topology changes frequently. Routing protocols therefore, play an important role in ad hoc network communications.

The most popular routing approach in ad hoc networks is on-demand routing. Instead of periodically exchanging route messages to maintain permanent route table of full topology, on-demand routing protocols build routes only when a node needs to send data packets to a destination. The source floods the packet to search the destination and discover the route. Link failures of only active routes are updated, and since no periodic route table exchange is required, control overhead is minimized and the routing information is utilized efficiently. Most proposed protocols of this type (for example, Dynamic Source Routing (DSR) [6] and Ad hoc On-demand Distance Vector (AODV) [14]) however, use single route for each session.

Multiple paths can be useful in improving the effective bandwidth of communication pairs, responding to congestion and bursty traffic, and increasing delivery reliability. In QoS routing in wired networks, multipath routing has been widely developed [2], [3], [10], [12], [17], [18], [20], [21]. These protocols use table-driven algorithms (link state or distance vector) to compute multiple routes. Studies show however, that proactive protocols perform poorly in mobile networks because of excessive routing overhead [1], [5]. Multipath routing in ad hoc networks has been proposed in [9], [11], [13], [15]. Although these protocols build multiple routes on demand, the traffic is not distributed into multipaths; only one route is primarily used and alternate paths are utilized only when the primary route is broken.

We present Split Multipath Routing (SMR) protocol that builds maximally disjoint paths. Multiple routes, of which one is the shortest delay path, are discovered on demand. Established routes are not necessarily of equal length. Data traffic is split into multiple routes to avoid congestion and to use network resources efficiently. We believe providing multiple routes is beneficial in network communications, particularly in mobile wireless networks where routes are disconnected frequently because of mobility and poor wireless link quality.

The remainder of this paper is organized as follows. Section II describes the protocol mechanism in detail. Performance evaluation by simulation is presented in Section III and concluding remarks are made in Section IV.

II. SPLIT MULTIPATH ROUTING

A. Route Discovery

Split Multipath Routing (SMR) is an on-demand routing protocol that builds multiple routes using request/reply cycles. When the source needs a route to the destination but no route information is known, it floods the ROUTE REQUEST (RREQ) message to the entire network. Because this packet is flooded, several duplicates that traversed through different routes reach the destination. The destination node selects multiple disjoint routes and sends ROUTE REPLY (RREP) packets back to the source via the chosen routes.

A.1 RREQ Propagation

The main goal of SMR is to build *maximally disjoint multiple paths*. We want to construct maximally disjoint routes to prevent

RED HAT 1012
RED HAT V. CAS

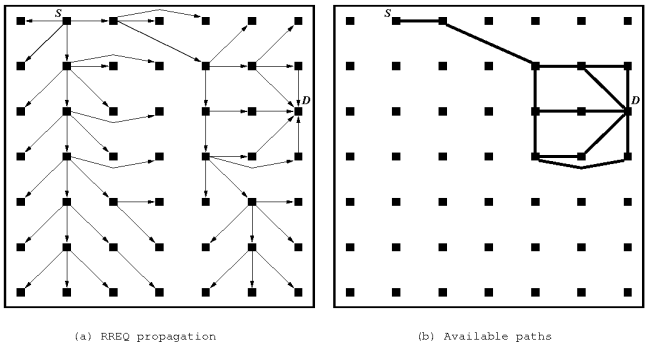


Fig. 1. Overlapped multiple routes.

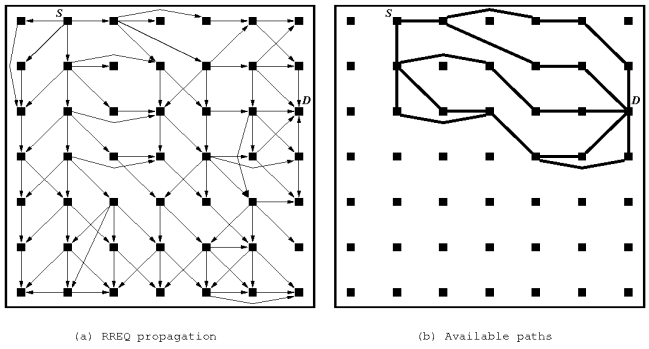


Fig. 2. Multiple routes with maximally disjoint paths.

certain nodes from being congested, and to utilize the available network resources efficiently. To achieve this goal in on-demand routing schemes, the destination must know the entire path of all available routes so that it can select the routes. Therefore, we use the source routing approach where the information of the nodes that consist the route is included in the RREQ packet. Additionally, intermediate nodes are not allowed to send RREPs back to the source even when they have route information to the destination. If nodes reply from cache as in DSR [6] and AODV [14], it is difficult to establish maximally disjoint multiple routes because not enough RREQ packets will reach the destination and the destination node will not know the information of the route that is formed from the cache of intermediate nodes.

When the source has data packets to send but does not have the route information to the destination, it transmits a RREQ packet. The packet contains the source ID and a sequence number that uniquely identify the packet. When a node other than the destination receives a RREQ that is not a duplicate, it appends its ID and re-broadcasts the packet. During simulation experiments however, we found out that dropping all duplicate RREQs only generate multiple paths that are mostly overlapped. Figure 1 (a) shows the paths taken by RREQs from the source node S to the destination node D, and Figure 1 (b) depicts the available routes. We can observe that all five routes share the first two links.

In order to avoid this overlapped route problem, we introduce a different packet forwarding approach. Instead of dropping every duplicate RREQs, intermediate nodes forward the duplicate packets that traversed through a different incoming link than the link from which the first RREQ is received, and whose hop count is not larger than that of the first received RREQ. Figure 2 (a) shows the paths taken by RREQs using this technique. We can select more disjoint paths from routes available in Figure 2 (b) than those in Figure 1 (b). Our approach has a disadvantage of transmiting more RREQ packets, but it enables us to discover maximally disjoint routes.

A.2 Route Selection Method

In our algorithm, the destination selects two routes that are maximally disjoint. More than two routes can be chosen, but we limit the number of routes to two in this study. One of the two routes is the shortest delay route; the path taken by the first RREQ the destination receives. We use the shortest delay path as one of the two routes to minimize the route acquisition latency required by on-demand routing protocols. When receiving the first RREQ, the destination records the entire path and sends a RREP to the source via this route. The node IDs of the entire path is recorded in the RREP, and hence the intermediate nodes can forward this packet using this information. After this process, the destination waits a certain duration of time to receive more RREQs and learn all possible routes. It then selects the route that is maximally disjoint to the route that is already replied. The maximally disjoint route can be selected because the destination knows the entire path information of the first route and all other candidate routes. If there are more than one route that are maximally disjoint with the first route, the one with the shortest hop distance is chosen. If there still remain multiple routes that meet the condition, the path that delivered the RREQ to the destination the quickest between them is selected. The destination then sends another RREP to the source via the second route selected. Note that two routes of the session are not necessarily of equal length.

Because our protocol uses the source routing and intermediate nodes do not reply from cache, only the source nodes maintain route information to destinations. Each node hence uses less memory, but packet header size is larger because we use source routing.

B. Route Maintenance

A link of a route can be disconnected because of mobility, congestion, and packet collisions. It is important to recover broken routes immediately to do effective routing. In SMR, when a node fails to deliver the data packet to the next hop of the route (by receiving a link layer feedback from IEEE 802.11 [4] or not receiving passive acknowledgments [7]), it considers the link to

RED HAT 1012
RED HAT V. CAS

be disconnected and sends a ROUTE ERROR (RERR) packet to the upstream direction of the route. The RERR message contains the route to the source, and the immediate upstream and downstream nodes of the broken link. Upon receiving this RERR packet, the source removes every entry in its route table that uses the broken link (regardless of the destination). If only one of the two routes of the session is invalidated, the source uses the remaining valid route to deliver data packets.

When the source is informed of a route disconnection and the session is still active, it may use one of the two policies in rediscovering routes:

- initiates the route recovery process when any route of the session is broken, or
- initiates the route recovery process only when both routes of the session are broken.

The first scheme reconstructs the routes more often and produces more control overhead than the second scheme, but the former provides multiple routes most of the time and be robust to route breaks. We evaluate both schemes by simulation in Section III.

C. Allocation Granularity

When the source receives a RREP after flooding the RREQ, it uses the first discovered route to send buffered data packets. When the second RREP is received, the source has two routes to the destination, and can split traffic into two routes. We use a simple *per-packet allocation* scheme when there are more than one available route to the destination. One drawback of this scheme is out of order delivery and re-sequencing burden on the destination (and hence not suitable to coexist with TCP). We believe, however, that cost-effective reordering buffers are easily implementable. We decided to use the per-packet allocation approach because it is known to work well in most networks [8], and most of all, it is fairly difficult to obtain the network condition (such as available bandwidth) in ad hoc networks to apply more sophisticated schemes.

III. PERFORMANCE EVALUATION

A. Simulation Environment

We evaluate and compare the performance of the following protocols:

- SMR-1: SMR which performs the route recovery when any route to the destination is invalidated
- SMR-2: SMR which performs the route recovery only when both routes to the destination are invalidated
- DSR: Dynamic Source Routing [6] which uses single path.

We implemented the simulator within the Global Mobile Simulation (GloMoSim) library [19]. Our simulation modeled a network of 50 mobile hosts placed randomly within a 1000 meter × 1000 meter area. Each node has a radio propagation range of 250 meters and channel capacity was 2 Mb/s. Each run executed for 300 seconds of simulation time.

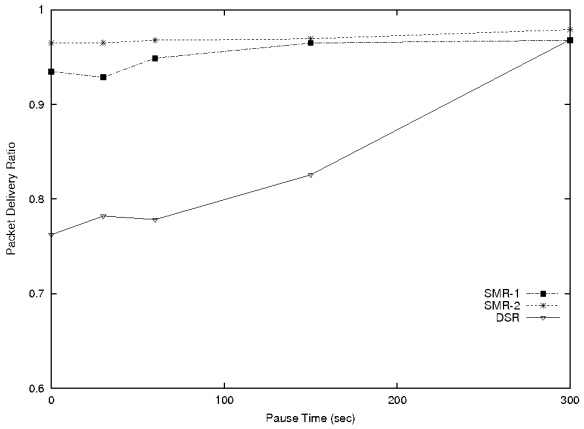


Fig. 3. Packet delivery ratio.

A free space propagation model with a threshold cutoff [16] was used in our experiments. In the radio model, we assumed the ability of a radio to lock onto a sufficiently strong signal in the presence of interfering signals, i.e., radio capture. We used the IEEE 802.11 Distributed Coordination Function (DCF) [4] as the medium access control protocol. A traffic generator was developed to simulate constant bit rate sources. There are twenty data sessions, and the sources and the destinations are randomly selected with uniform probabilities. The size of data payload was 512 bytes. We used random waypoint model [6] as the mobility model. We generated various mobility degree by using different pause times. The minimum and the maximum speed were set constant to zero and 10 m/s, respectively.

B. Results and Analysis

Figure 3 shows the throughput of each protocol in packet delivery fraction. Packet delivery ratio is obtained by dividing the number of data packets correctly received by the destinations by the number of data packets originated by the sources. We can observe from the result that both SMR schemes outperform DSR, especially when the mobility increases (i.e., the pause time decreases). In DSR, only one route is used for each session and when that route is invalidated, the source uses the cached route that is learned from overhearing packets. If no such cached route is available, it sends a RREQ to discover a new route. In the latter case, intermediate nodes that have cached routes to the destination provide those route to the source by sending RREPs. DSR however, does not apply any aging mechanism for cached route entries, and hence routes stored in the cache (either by the source or the intermediate nodes) may be stale. After a route break, source nodes will use these newly acquired but obsolete routes only to learn that they are also invalid, and will attempt another route recovery. Many data packets are dropped during this process and more delay is needed to discover correct routes.

RED HAT 1012
RED HAT V. CAS

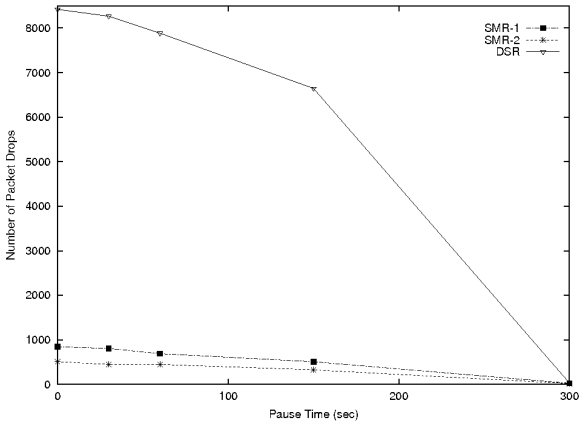


Fig. 4. Number of packet drops.

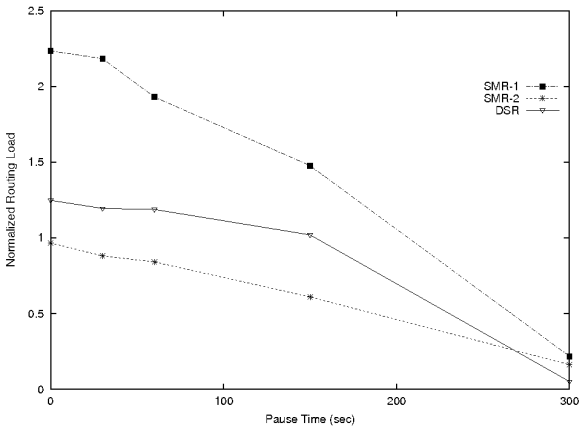


Fig. 5. Normalized routing load.

Between SMR protocols, SMR-2 delivers more packets than SMR-1. We can analyze that the control packets generated by the route rediscovery processes of SMR-1 cause collision and contention with data packets. Even though SMR-2 will have only one available route to the destination after the other route is broken, it can still deliver data packets without producing control traffic as long as the remaining route stays connected, and that leads to a good throughput performance.

Figure 4 illustrates the number of packets dropped by each protocol. Both data and control packets are measured. The reasons for packet drops can be incorrect route information, mobility, collisions, and congestion. DSR cannot maintain precise routes and drops more packets as nodes move more often (i.e., less pause time). The usage of state routes from caches is the major reason of DSR packet drops. Both SMR schemes have considerably fewer packet drops compared with DSR. SMR-2 has fewer packet drops than SMR-1 because it invokes fewer route recovery processes and consequently, transmits less control messages.

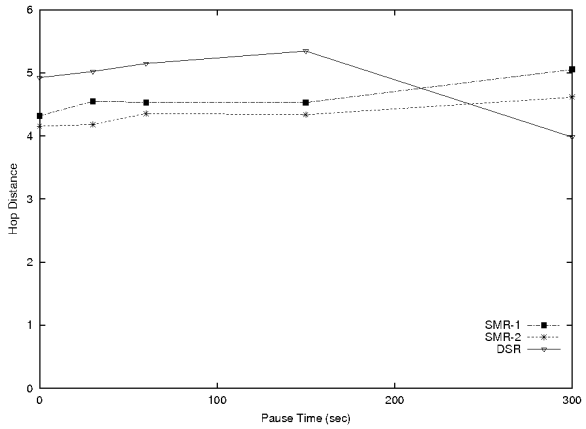


Fig. 6. Hop distance.

Figure 5 presents the control overhead in normalized routing load. Normalized routing load is the ratio of the number of control packets propagated by every node in the network and the number of data packets received by the destination nodes. This value hence represents the protocol efficiency. When there is no mobility, DSR has the least value. This result is expected because SMR protocols generate more control packets while building multiple routes. On the other hand, DSR builds single route for each session and minimizes flooding overhead by allowing intermediate nodes of replying from cache. Cached routes are useful in static networks as they remain valid for the entire duration. As mobility is increased, however, SMR-2 shows better efficiency than DSR. DSR yields less overhead in initial route discovery process, but it invokes more route reconstruction procedures than SMR-2 since DSR intermediate nodes often reply with stale routes. Additionally, DSR transmits considerably more RERR packets than SMR schemes because DSR has more route disconnections and route recoveries. Furthermore, DSR sends RERR packets whenever a unicast packet (data, RREP, and RERR) fails to be delivered to the next hop. SMR sends RERR only when the data packet is undeliverable. Therefore, DSR shows higher normalized routing load than SMR-2 when mobility is present. We can also observe that SMR-1 shows less efficiency than other protocols regardless of mobility. Since the source floods the network with RREQs when any route of a session is disconnected, more control packets are transmitted than DSR and SMR-2. We can deduce from this result that excessive flooding makes the protocol inefficient.

Figure 6 reports the average hop distance of each protocol. DSR has the shortest hop distance when there is no mobility because SMR schemes' second routes may have longer distance than the first routes. With mobility however, the hop distance of DSR grows and becomes larger than those of SMR protocols. If the route is established directly from the destination, it can be the shortest route since it is built based on the most recent information and accounts for node locations after movements. DSR, however, uses cached routes from intermediate nodes. These

RED HAT 1012
RED HAT V. CAS

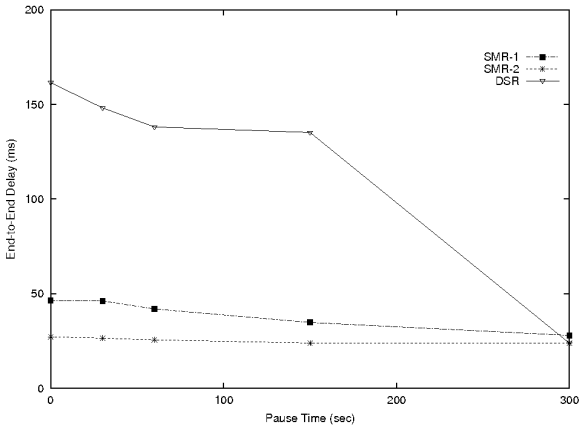


Fig. 7. End-to-end delay.

routes may not be fresh and do not exploit the current network topology. DSR therefore builds longer routes than SMR protocols. Longer paths have a more chance of having route breaks since one link disconnection results in a route invalidation. Results from Figure 3 confirms our observation.

Figure 7 shows the end-to-end delay. DSR has the longest delay in mobile scenarios because it delivers data packets on routes longer than those of SMR. In addition, DSR yields longer delays in reconstructing routes and the period of time the data packets are buffered at the source node during route recovery results in larger end-to-end delays. SMR on the other hand, uses the remaining valid route when one of the multiple route is disconnected, and hence no route acquisition latency is required.

IV. CONCLUSION

We presented the Split Multipath Routing (SMR) protocol for ad hoc networks. SMR is an on-demand protocol that builds maximally disjoint routes. Our scheme uses two routes for each session; the shortest delay route and the one that is maximally disjoint with the shortest delay route. We attempt to build maximally disjoint routes to avoid having certain links from being congested, and to efficiently utilize the available network resources. Providing multiple paths is useful in ad hoc networks because when one of the route is disconnected, the source can simply use other available routes without performing the route recovery process.

We introduced two approaches in SMR route maintenance. The first scheme builds a new pair of routes when any existing route of the session is disconnected. The second scheme performs rerouting only when both routes are broken. We have conducted a simulation performance evaluation of these two SMR schemes and DSR which uses single shortest delay route. Our study indicates that SMR outperforms DSR because multiple routes provide robustness to mobility. The performance difference becomes evident as the mobility degree increases. SMR had considerably fewer packet drops compared with DSR. Splitting the traffic into multiple routes helps distribute the load to the network hosts. SMR also showed shorter end-to-end delay because route acquisition latency is not required for all route disconnections. Between SMR protocols, the second scheme showed better efficiency as it performs fewer route recoveries and hence generates less control overhead.

References

[1] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proceedings of ACM/IEEE MOBICOM'98*, Dallas, TX, Oct. 1998, pp. 85-97.

[2] J. Chen, P. Druschel, and D. Subramanian, "An Efficient Multipath Forwarding Method," *Proceedings of IEEE INFOCOM'98*, San Francisco, CA, Mar. 1998, pp. 1418-1425.

[3] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of Multi-Path Routing," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, Dec. 1999, pp. 885-896.

[4] IEEE Computer Society LAN MAN Standards Committee, *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification*, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.

[5] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks," *Proceedings of ACM/IEEE MOBICOM'99*, Seattle, WA, Aug. 1999, pp. 195-206.

[6] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Chapter 5, Kluwer Academic Publishers, 1996, pp. 153-181.

[7] J. Jubin and J.D. Tornow "The DARPA Packet Radio Network Protocols," *Proceedings of the IEEE*, vol. 75, no. 1, Jan. 1987, pp. 21-32.

[8] R. Krishnan and J.A. Silvester, "Choice of Allocation Granularity in Multipath Source Routing Schemes," *Proceedings of IEEE INFOCOM'93*, San Francisco, CA, Mar. 1993, pp. 322-329.

[9] S.-J. Lee and M. Gerla, "AODV-BR: Backup Routing in Ad hoc Networks," *Proceedings of IEEE WCNC 2000*, Chicago, IL, Sep. 2000.

[10] S. Murthy and J.J. Garcia-Luna-Aceves, "Congestion-Oriented Shortest Multipath Routing," *Proceedings of IEEE INFOCOM'96*, San Francisco, CA, Mar. 1996, pp. 1028-1036.

[11] A. Nasipuri and S.R. Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks," *Proceedings of IEEE ICCCN'99*, Boston, MA, Oct. 1999, pp. 64-70.

[12] R. Ogier, V. Rutenburg, and N. Shacham, "Distributed Algorithms for Computing Shortest Pairs of Disjoint Paths," *IEEE Transactions on Information Theory*, vol. 39, no. 2, Mar. 1993, pp. 443-455.

[13] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of IEEE INFOCOM'97*, Kobe, Japan, Apr. 1997, pp. 1405-1413.

[14] C.E. Perkins and E.M. Royer, "Ad-Hoc On Demand Distance Vector Routing," *Proceedings of IEEE WMCSA'99*, New Orleans, LA, Feb. 1999, pp. 90-100.

[15] J. Raju and J.J. Garcia-Luna-Aceves, "A New Approach to On-demand Loop-Free Multipath Routing," *Proceedings of IEEE ICCCN'99*, Boston, MA, Oct. 1999, pp. 522-527.

[16] T.S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, Oct. 1995.

[17] D. Sidhu, R. Nair, and S. Abdallah, "Finding Disjoint Paths in Networks," *Proceedings of ACM SIGCOMM'91*, Zurich, Switzerland, Sep. 1991, pp. 43-51.

[18] N. Taft-Plotkin, B. Bellur, and R. Ogier, "Quality-of-Service Routing Using Maximally Disjoint Paths," *Proceedings of IEEE IWQoS'99*, London, UK, Jun. 1999, pp. 119-128.

[19] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, *GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems*, `http://pcl.cs.ucla.edu/projects/domains/glomosim.html`.

[20] S. Vutukury and J.J. Garcia-Luna-Aceves, "An Algorithm for Multipath Computation Using Distance-Vectors with Predecessor Information," *Proceedings of IEEE ICCCN'99*, Boston, MA, Oct. 1999, pp. 534-539.

[21] W.T. Zaumen and J.J. Garcia-Luna-Aceves, "Loop-Free Multipath Routing Using Generalized Diffusing Computations," *Proceedings of IEEE INFOCOM'98*, San Francisco, CA, Mar. 1998, pp. 1408-1417.

RED HAT 1012
RED HAT V. CAS

Improving TCP Performance in Multipath Packet Forwarding Networks

Youngseok Lee, Ilkyu Park, and Yanghee Choi

Abstract: **This paper investigates schemes to improve TCP performance in multipath forwarding networks. In multipath routing, packets to the same destination are sent to multiple next-hops in either packet-level or flow-level forwarding mode. Effective bandwidth is increased since we can utilize unused capacity of multiple paths to the destination. In packet-level multipath forwarding networks, TCP performance may not be enhanced due to frequent out-of-order segment arrivals at the receiver because of different delays among paths. To overcome this problem, we propose simple TCP modifications. At the sender, the fast retransmission threshold is adjusted taking the number of paths into consideration. At the receiver, the delayed acknowledgment scheme is modified such that an acknowledgment for an out-of-order segment arrival is delayed in the same way for the in-order one. The number of unnecessary retransmissions and congestion window reductions is diminished, which is verified by extensive simulations. In flow-level multipath forwarding networks, hashing is used at routers to select outgoing link of a packet. Here, we show by simulations that TCP performance is increased in proportion to the number of paths regardless of delay differences.**

Index Terms: **TCP, fast retransmission, delayed ACK, multipath.**

I. INTRODUCTION

A. Multipath Routing

Multipath routing is employed in order to increase the total network utilization and the end-to-end throughput. The inter-domain routing protocols such as Border Gateway Protocol (BGP) [1] in the Internet are usually based on policy, while most intra-domain routing protocols such as Open Shortest Path Forwarding (OSPF) [2] and Intermediate System to Intermediate System (IS-IS) [3] adopt the simple shortest path routing algorithm which basically uses a single path from a source to a destination. Although the shortest path algorithm is simple and can easily be implemented, network resources may not be efficiently utilized. For example, connections along a congested path will experience severe packet delays or losses while there exist alternate unloaded paths between the source and the destination. In this case, congestion at the bottleneck link or path will be reduced through the increased bandwidth by multiple paths, which improves end-to-end performance. Besides, multipath routing efficiently utilizes the network by balancing the load among redundant paths.

The multipath routing model is defined as a routing model where routing algorithms provide potentially multiple paths between node pairs. Essential components to make multipath networks viable are algorithms which can compute multiple paths, an efficient multipath forwarding method, and a multipath transport protocol.

Routers can provide multiple paths with recently developed or proposed routing protocols. The easiest extension to multipath routing is to use the equal-cost multiple shortest paths when calculating the shortest one, which is known as Equal-Cost Multi-Path (ECMP) routing [2]. This is explicitly supported by several routing protocols such as OSPF and IS-IS. Some router implementations allow equal-cost multipath with Routing Information Protocol (RIP) or other routing protocols [4]. Recently, a multipath extension to the distance vector and link state routing protocols is added in [5], [6].

In the Multi-Protocol Label Switching (MPLS) [7] network, where IP datagrams are switched by looking up the fixed-size label, paths between an ingress router and an egress router can be explicitly set up by a signaling protocol, Constraint-based Routed Label Distribution Protocol (CR-LDP) or the modified Resource ReSerVation Protocol (RSVP). Therefore, multiple Label Switched Paths (LSPs) between an ingress router and an egress router allow the network to utilize the alternate non-shortest path for multipath routing.

A router employing multipath routing protocol distributes incoming packets with the same Forwarding Equivalent Class (FEC) to multiple outgoing links or next-hops. Routers forward packets either in packet-level or flow[1]-level mode (Fig. 1).

In the packet-level forwarding method, packets for the same FEC are delivered to one of multiple next-hops in round-robin or random manner. Therefore, packets belonging to the same flow are forwarded to different next-hops. Although this mechanism can be easily implemented, it will cause many out-of-order packet arrivals at the destination since multiple paths may have different delays. In order to remedy possible end-to-end performance degradation, an additional buffer management scheme to fix the mis-ordering packets is required. On the contrary, flow-level forwarding routers deliver packets belonging to a flow along the same next-hop. Routers classify packet streams into flows using flow detectors such as the X/Y (X: the number of packets, Y: timeout) flow classifier of the IP switch [8] or the NetFlow of the Cisco routers [9]. The flow detectors will

Manuscript received February 23, 2001; approved for publication by Danny Raz, Division III Editor, January 8, 2002.

Y. Lee is with Multimedia and Computer Communication Laboratory, School of Computer Science and Engineering, Seoul National University, Seoul, Korea, e-mail: yslee@mmlab.snu.ac.kr.

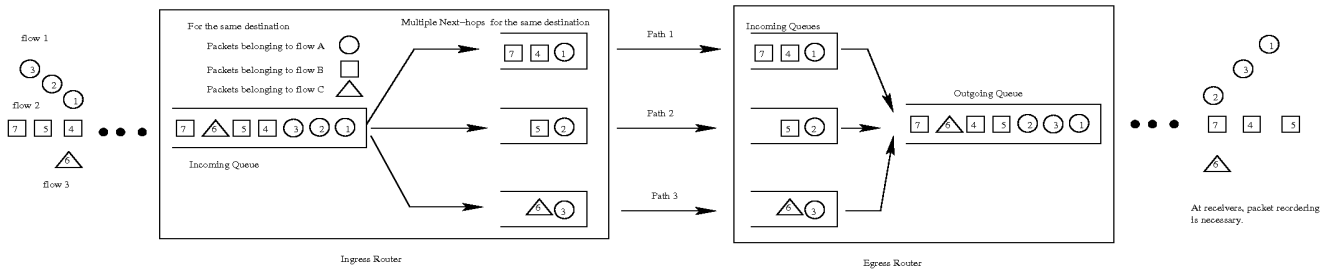
I. Park is with Distributed Virtual Reality Research Team, Electronics and Telecommunications Research Institute (ETRI), Taejon, Korea, e-mail: xiao@etri.re.kr.

Y. Choi is with Multimedia and Computer Communication Laboratory, School of Computer Science and Engineering, Seoul National University, Seoul, Korea, e-mail: yhchoi@mmlab.snu.ac.kr.

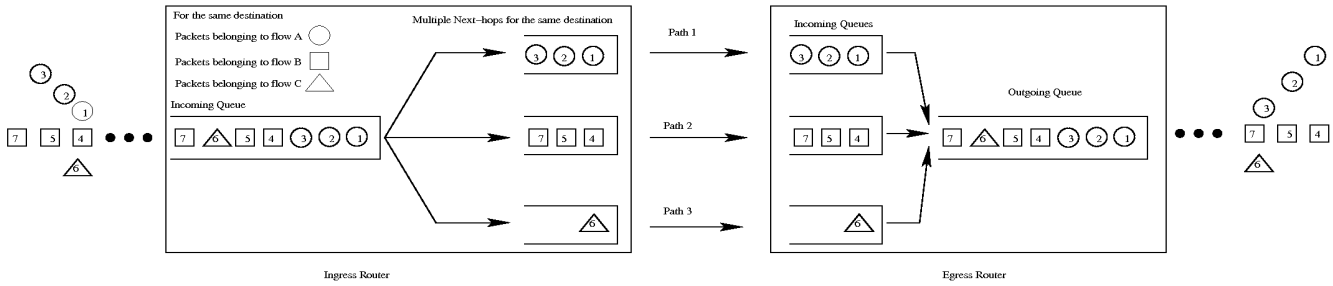
[1] In this paper, a flow is defined by packet stream belonging to the same flow specification consisting of the end-point addresses (e.g., source/destination IP addresses and port number) and a timeout between consecutive packet arrivals.

1229-2370/02/$10.00 © 2002 KICS

RED HAT 1012
RED HAT V. CAS



(a) Packet-level multipath forwarding



(b) Flow-level multipath forwarding

Fig. 1. Multipath packet forwarding schemes in routers: In packet-level forwarding, packet arrivals may be out-of-order, while packet sequence is kept in flow-level forwarding.

consider packet streams belonging to the same flow specification as a flow when X packets arrive within Y timeout. That is, if the number of packet arrivals for the given flow specification is less than X within Y timeout, the flow will not be detected or will be deleted from the flow table. However, these schemes require per-flow states in routers, thus not scalable. A hashing function can be used for stateless multipath forwarding [10], and load balancing is achieved by adjusting the hashing range. Yet, it is required that implementation of the hashing function is deployed to all the routers supporting multipath routing. With multiple paths at the network layer, the traditional transport protocol will not perform well. Instead of devising a new multipath transport protocol [11], we propose a simple modification of the existing transport protocol, namely TCP.

B. TCP Performance over Multiple Paths

Transmission Control Protocol (TCP) provides reliable data transfer between node pairs. TCP receivers send an immediate acknowledgment (ACK) when a segment arrives. These ACKs are cumulative and acknowledge all in-order segment arrivals. TCP senders detect a packet loss by a timer expiration or duplicate ACKs. In order to recover the lost packet fast, TCP senders retransmit the lost packet when three duplicate ACKs arrive (*the fast retransmit* algorithm). After the fast retransmission, the congestion avoidance is performed (*the fast recovery* algorithm). An optional *delayed acknowledgment* mechanism [12] allows TCP receivers to send an ACK after a timer expiration (traditionally 200 ms[2]) or at least at every second full-sized segment arrivals. However, when out-of-order segments arrive at a receiver, receivers send duplicate ACKs for each packet to recover the lost packet. Delayed ACKs decrease the amount of data which senders can send because the number of ACKs is reduced, and they are harmful during slow start [13]. However, during the congestion avoidance period, delayed ACKs will conserve network bandwidth and host computation resources by sending fewer ACKs without drastically reducing performance. Delayed ACKs are useful for bulk transfers and asymmetric links [14].

TCP performance over multiple paths is supposed to be enhanced, because the bottleneck bandwidth increases. However, TCP senders will regard triple duplicate ACKs, resulting from out-of-order packet arrivals at the receiver due to the different delays of multiple paths, as an indicator of a packet loss, and will retransmit the lost packet (Fast Retransmission). Not only this wastes extra bandwidth, but also it reduces the congestion window (*cwnd*) of the TCP sender (Fast Recovery). Moreover, if ACKs arrive at the TCP sender out-of-order, the late segment with the lower sequence number will be ignored and cannot serve for increasing the *cwnd*, since TCP senders accept only the in-order ACKs. Besides, the variable path Maximum Transfer Unit (MTU) size for each packet will make path MTU discovery useless. The loss rate (p_i) and the Round-Trip Time (RTT_i) over N multiple paths will degrade TCP performance by $\frac{1}{\sqrt{\sum_i^N p_i}}$ and $\frac{1}{RTT_i}$ [15]. Hence, TCP performance in packet-level multipath forwarding may not be increased under the conventional environment.

This paper proposes simple TCP modification schemes to prevent end-to-end performance degradation in packet-level multipath forwarding. Then, it is verified that the hashing-based

[2]The delay should be less than 500 ms [12], and BSD-based TCP implementations use 200 ms usually.

RED HAT 1012
RED HAT V. CAS

flow-level multipath forwarding method linearly increases linearly TCP performance with no need of per-flow states.

The remainder of this paper is organized as follows. Related works are introduced in Section II. In Section III, we propose TCP modifications and verify the effectiveness of the hashing-based flow mapping scheme. The results of performance evaluation by simulation are discussed in Section IV, and Section V concludes this paper.

II. RELATED WORKS

In connection-oriented networks, Cidon *et al.* have analyzed the performance of multipath routing algorithms and have shown that the connection establishment time for multipath reservation is significantly lowered [16]. Park and Zarki have proposed a dynamic multipath routing algorithm in connection-oriented networks, where the shortest path is used under light traffic condition and multiple paths are utilized as the shortest path becomes congested [17]. Therefore, in connection-oriented networks, only connection or call-level routing and forwarding are considered. Quality-of-Service (QoS) routing via multiple paths under time constraint is proposed when the bandwidth can be reserved, assuming all the reordered packets are recovered by the optimal buffer at the receiver, with much overhead of the dynamic buffer adjustment at the receiver [18]. The enhanced routing scheme for load balancing by separating long-lived and short-lived flows is proposed and it is shown that congestion can be greatly reduced [19]. It is shown that the quality of services can be enhanced by dividing the transport-level flows into UDP and TCP flows [20]. Yet, it does not consider the aggregated flows. In the MPLS network, a traffic engineering method using multiple multipoint-to-point LSPs is proposed [21]. As backup routes are used against failures, the alternate paths are used only when primary routes do not work.

Although there have been researches on multipath routing algorithms, the effects on the transport or application layers are largely ignored. Recently, a reliable bit stream multipath transport protocol, MPTCP, is proposed for multipath forwarding networks [11]. Although the MPTCP improves the TCP throughput over multiple paths, it is required that multiple independent TCP connections are established. Therefore, in MPTCP, a sender's data stream is fragmented and attached with two-level (MPTCP and TCP) sequence numbers, and the original data stream is reconstructed at the receiver. Although TCP designed for a single path is naturally extended for the multiple paths, the implementation overhead of the end-host is high.

For the improvement of TCP performance, several modified TCP acknowledgments are proposed. Allman has proposed a byte counting method to improve TCP performance degradation during/after slow start when using delayed ACKs [13]. In asymmetric networks, Balakrishnan *et al.* maintain that TCP performance depends on forward path as well as backward path, and propose dynamically varying delayed ACKs [14]. The effects of extended acknowledgment interval on TCP performance are analyzed, and it is shown that extending ACK interval increases the TCP throughput when two machines are on relatively close networks [22].

III. TWO APPROACHES TO IMPROVE TCP PERFORMANCE OVER MULTIPLE PATHS

This section explains two approaches to enhance TCP performance over the multiple paths according to multipath forwarding capabilities of routers.

A. End-host Modifications in Packet-level Multipath Forwarding Networks

When routers support only packet-level multipath forwarding, minor TCP modifications are enough to enhance end-to-end performance. The proposed schemes are either a sender-side solution, a receiver-side one, or an integrated scheme that both the sender and the receiver use TCP modification.

A.1 Sender-side Solution: Increased Fast Retransmission Threshold

The default fast retransmission threshold value (= 3) of the TCP senders may not be suitable where in packet-level multipath forwarding networks many out-of-order packet arrivals occur due to delay mismatches among path. At the sender, duplicate ACKs are considered as an indication of packet loss, and the lost segment is retransmitted after triple duplicate ACK arrivals. By increasing the fast retransmission threshold unnecessary retransmissions resulting from out-of-order segment arrivals can be avoided. When long-lived flows are multicast from a sender (or server) to a lot of receivers (or clients) along multiple paths usual in multimedia applications, this solution is very effective because no change is needed at the receivers.

We propose a simple heuristic for determining the fast retransmission threshold, T, under the given number of multiple paths, N. Without loss of generality, the number of practical multiple links or paths will be restricted to a small value (e.g., 2 - 8), because multiple next-hops for the same destination correspond to interfaces at a router. To take advantage of practical multiple paths, we classify multiple paths into the range of $[2^{N-1}, 2^N - 1]$, which may be thought as a small multipath set consisting of two or three paths, a middle-sized multipath set of 4 - 7 paths, a large one with 8 - 15 paths, and the huge set with over 16 paths. Then, for each multipath range, we increase the fast retransmission threshold by the number of duplicate ACKs, D, which is set to three for a single path by default, as follows.

$$T = D \cdot (1 + \lfloor \log_2 N \rfloor) \quad (1)$$

Fig. 2 shows the fast retransmission threshold as the number of multiple paths increases. When two or three paths are used, TCP enters the fast retransmission mode after six duplicate ACK arrivals. Therefore, receivers have more time to wait for late packet arrivals through different paths. Determining the fast retransmission threshold is validated by simulation in the next section.

A.2 Receiver-side Solution: Modified Delayed ACKs

When it is difficult to change TCP senders, TCP receivers can employ modified delayed ACKs to achieve TCP performance improvement as follows. Therefore, we propose the following

RED HAT 1012
RED HAT V. CAS

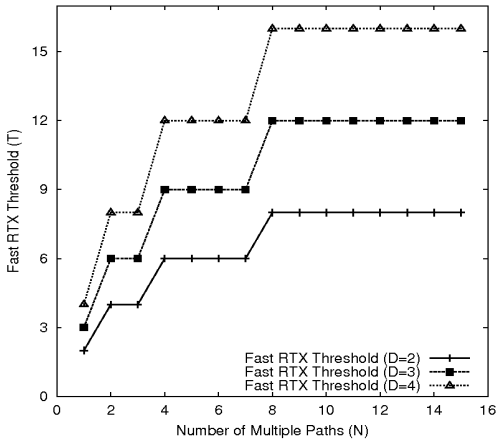


Fig. 2. Fast retransmission threshold according to the number of multiple paths.

modification of delayed ACKs for TCP receivers to reduce duplicate ACKs.

1. **Delaying ACKs for out-of-order segments**: TCP receivers do not send an ACK immediately for each out-of-order packet. Instead, an ACK is generated at every two segment arrivals or after a delayed ACK timer expiration regardless of packet sequence orders.
2. **Sending immediately ACKs for retransmitted segments**: If ACKs for out-of-order segments are delayed, it is possible that two or more retransmission events occur. For instance, under the round trip time less than the traditional delayed ACK timer (200 ms), although the sender retransmits the lost packet, TCP receivers employing the proposed delayed ACK may wait until two segments arrive or the timer expires, which causes a retransmission timeout. Therefore, receivers should send an ACK immediately for a retransmitted segment during the duplicate ACK period.

With the proposed delayed ACK method, the probability that the window reduction resulting from triple duplicate ACKs by out-of-order packet arrivals will be decreased, thereby avoiding TCP performance degradation.

In general, delayed ACKs may decrease the TCP throughput by $\frac{1}{\sqrt{b}}$ (b = number of packets per ACK) [15] in a single path. In addition, receivers employing modified delayed ACKs respond slowly to a packet loss because ACKs for out-of-order packets are delayed. However, when packet arrivals are frequently out-of-order in packet-level multipath forwarding networks, TCP performance will be enhanced with modified delayed ACKs. Furthermore, by reducing the number of ACKs, the receiver-side solution is useful in the asymmetric network where up link bandwidth is scarce.

A.3 Integrated Solution: Increased Fast Retransmission Threshold and Modified Delayed ACKs

When both the sender and the receiver use the modified TCP, the probability that the mis-ordered TCP segments will be recovered at the receiver increases in multipath routing. In the single shortest path-based routing, however, TCP throughput will be degraded, because it takes longer for the sender to detect the real segment loss while the receiver delays ACKs and the sender waits for additional duplicate ACKs.

B. Router-based Approach: Flow-level Multipath Forwarding

When forwarding is done in flow by flow, routers send packets belonging to the same TCP flow to the same next-hop. Therefore, out-of-order packet arrivals will be prohibited in advance. End hosts are not changed and at routers we use the hashing-based flow mapping scheme [10] that does not need flow states. Flow-aware routers generally keep per-flow states in memory to detect a flow and map it into an appropriate path. However, the hashing-based method removes the scalability problem with the stateless operation, which becomes suitable in high-speed networks. Although it is required that the additional implementation of the hashing function should be deployed to all the multipath-capable routers, end-to-end performance as well as the network resource utilization will be increased in proportion to the number of paths without the overhead of the end host modification.

IV. PERFORMANCE EVALUATION

A. Effects of Increased Fast Retransmission Threshold

We used the ns network simulator [23] for our experiments. The network topology is shown in Fig. 3, where up to five paths can be used between a source and a destination.

All the access links ($S_i - R_0, R_1 - D_i$) have bandwidths of 100 Mbps and link delays of 1 ms, while $R_i - R_j$ links in the core network have longer delays (10 ms) with identical bandwidth (100 Mbps). For each node pair (S_i, D_i), ten FTP/TCP (NewReno) connections are opened. Therefore, in total, one hundred FTP connections are run. The TCP senders use the segment size of 512 bytes, and the advertised window size is 100. For the background traffic, ten Pareto connections (one for each node) compete for the bottleneck links between R_0 and R_1. The Pareto traffic sources are configured with the packet size of 500 byte, burst time of 500 ms, idle time of 500 ms, and shape value of 1.5. The peak rate of each Pareto source is 10 Mbps, which will cause congestion in case of a single path between R_0 and R_1. To examine the effects of multipath forwarding, up to five paths can be used between R_0 and R_1. The router adopts the drop-tail queue discipline for simplicity.

For the comparison of TCP performance, we use the relative TCP goodput which is computed as the total received TCP segments except retransmission of the proposed scheme over that of the single path case of the standard TCP sender and receiver. It is shown in Fig. 4 that, when the link delays are different (100 %) and link loss is 0.001, TCP performance is generally enhanced as the fast retransmission threshold increases in multipath routing; however, TCP performance in single path routing begins to degrade when the fast retransmission threshold is greater than six. Under two or three multiple paths, it is shown that TCP performance is not much enhanced after six or more duplicate ACKs. In case of four or five multiple paths, increasing the

RED HAT 1012
RED HAT V. CAS

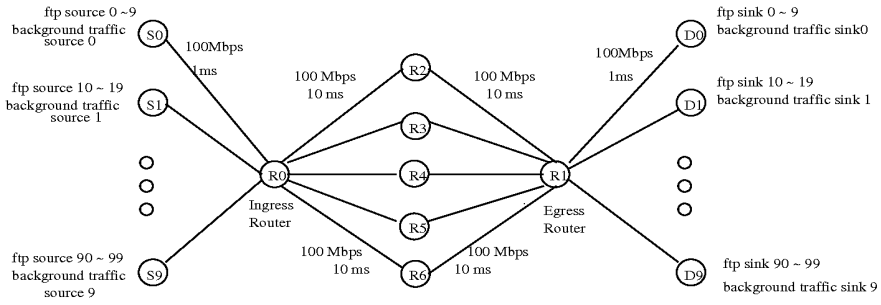


Fig. 3. Network topology for simulation.

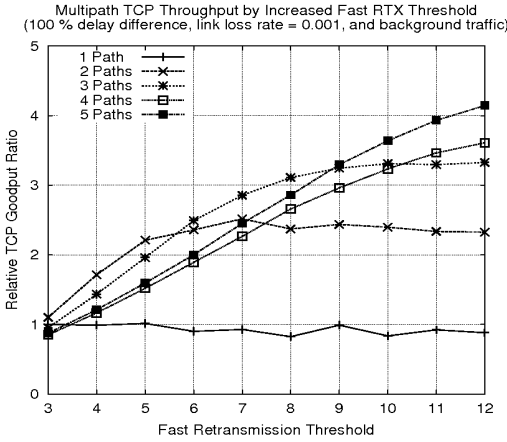


Fig. 4. The effects of changing the fast retransmission threshold.

fast retransmission threshold continuously enhances TCP performance. Yet, changing the threshold affects TCP performance differently according to the number of multiple paths. For instance, with the threshold of six, TCP performance of two or three paths shows better than that of four or five paths, while with the threshold of nine, TCP performance of four or five paths is better than that of two or three paths. Therefore, we use the heuristic for determining the fast retransmission threshold for the number of multiple paths in Eq.(1).

B. Effects of the Number of Multiple Paths with Different Delays and Loss Rates

In the same simulation environment (Fig. 3), we investigate the effects of the number of paths with different delays and link loss rates.

Fig. 5 shows the TCP throughput as the fast retransmission threshold is increased under the different path delays and link loss rate. The relative TCP goodput ratio is the TCP goodput of multiple paths over TCP goodput of a single path. TCP performance increases linearly. TCP senders with the default fast retransmission threshold (3) do not benefit from the increased bandwidth, and TCP performance is even lower than that of the single path. On the other hand, TCP performance of the proposed fast retransmission threshold method is drastically improved regardless of different path delays.

Especially, in case of five paths, TCP performance of 25 % path delay difference outperforms that of the same path delays, because most out-of-order packet arrivals are recovered and packet drops are reduced by the increased *delay-bandwidth* product. However, as the difference in path delays becomes large, many out-of-order packet arrivals degrade TCP performance. Still, under lossy links, TCP throughput increases as the number of paths by increasing the fast retransmission threshold.

Fig. 6 depicts the relative TCP throughput ratio by proposed delayed ACKs. As the path delay difference becomes large, TCP performance by the non-delayed or the standard delayed ACKs is severely degraded, even lower than that of a single path, because many out-of-order TCP segment arrivals are considered as packet losses. When receivers adopt modified delayed ACKs (Fig. 6-(c)), TCP performance is improved. In case of two or three paths, the modified delayed ACK method increases the TCP throughput more than two times in spite of the different path delays. However, when the number of paths is four or five, TCP performance is not any more improved by the modified delayed ACK. Fig. 7 shows the relative TCP throughput ratio when link loss rates are added. Under link loss rate of 0.001, TCP throughput by modified delayed ACK is enhanced. Therefore, we conclude that when routers support only packet-level multipath forwarding, two or three paths are enough for modified delayed ACKs.

In Fig. 8, TCP performance is shown when both the sender and the receiver use the modified TCP. The integrated scheme enhances TCP performance more than the sender-only or receiver-only method, because the packet recovery probability increases.

Next, we replace packet-level forwarding routers with flow-level packet forwarding ones. Each TCP flow is mapped to one of multiple next-hops by hashing the source address. As shown in Fig. 9, the TCP throughput of multiple paths in a flow-level forwarding network increases in proportion to the number of paths irrespective of the different delays and link loss, because packet sequences are maintained for every TCP flow along each path.

C. Effects of Partial and Explicit Multiple Paths

The explicit path setup of the MPLS network provides increased TCP performance when combined with flow-level multipath forwarding. Through the explicit Label Switched Paths

RED HAT 1012
RED HAT V. CAS

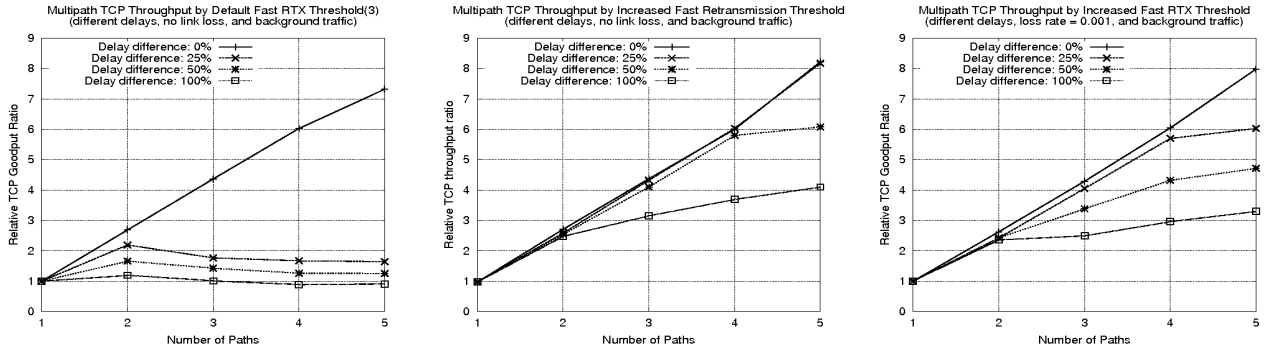


Fig. 5. TCP performance according to the fast retransmission threshold in a packet-level forwarding network: the fast retransmission threshold is 6 and 9 for 2-3 and 4-5 multiple paths, respectively.

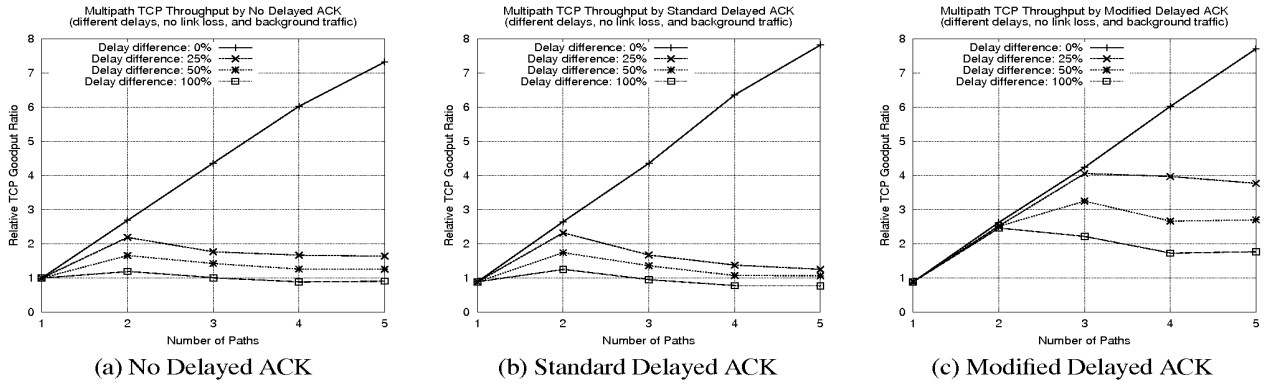


Fig. 6. TCP performance according to the types of TCP receivers in a packet-level forwarding network.

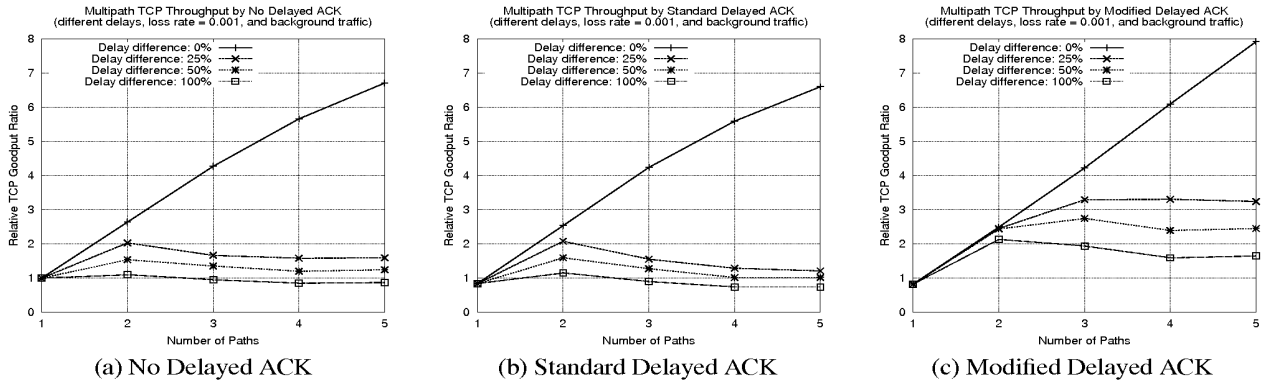
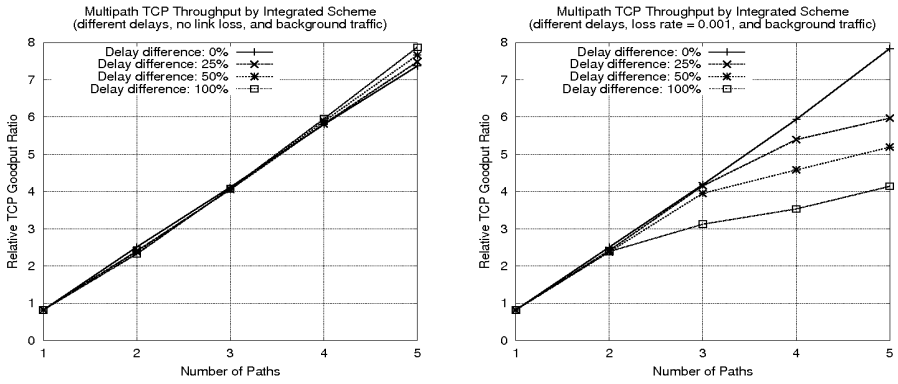


Fig. 7. TCP performance according to the types of TCP receivers in a packet-level forwarding network with lossy links.

(LSPs), the non-shortest paths can be efficiently utilized for multipath routing. Therefore, when flow-level forwarding routers provide explicit multiple paths instead of equal cost multipath, both the network resource utilization and the end-to-end throughput can be maximized.

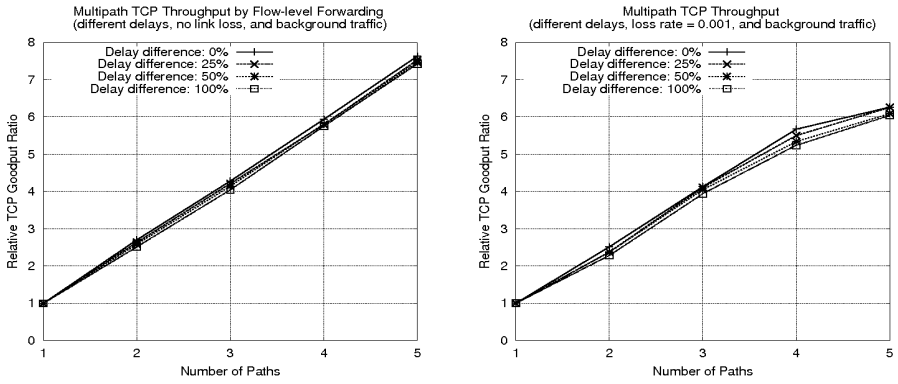
We examine a more complex network (Fig. 10) to see the effects of partial and explicit multiple paths. In this network topology (Fig. 10-(a)), when a single shortest path is used between R_0 and R_1, the bottleneck bandwidth between senders and receivers would be 100 Mbps of the $R_2 - R_4 - R_1$ path. If ECMP is supported, two equal-cost multiple paths ($R_0 - R_2 - R_3 - R_1$, and $R_0 - R_2 - R_4 - R_1$) increase the bottleneck bandwidth

RED HAT 1012
RED HAT V. CAS



(a) Without link loss (b) With link loss

Fig. 8. TCP performance by the integrated scheme that both the sender and the receiver use the modified TCP.



(a) Without link loss (b) With link loss

Fig. 9. TCP performance in a flow-level forwarding network.

to 155 Mbps of the $R_0 - R_2$ link, and the background traffic will be divided into two paths by half. In the MPLS network, however, three explicit paths (the above two, and non-shortest $R_0 - R_5 - R_6 - R_7 - R_1$) can be set up and used for multipath forwarding, thereby increasing the bottleneck bandwidth from 155 Mbps to 255 Mbps.

Under the same FTP/TCP configuration to the previous simulation, the average TCP goodput of one hundred FTP flows is shown in Fig. 10-(b). In multipath routing with ECMP support, modified delayed ACKs and integrated scheme enhance TCP performance by 80 %, when compared to single path case. Also, TCP performance of the fast retransmission threshold of six is improved by 94 %. In the flow-level forwarding MPLS network with two (three) LSPs, TCP performance is 2.1 (3.8) times more than that of single path.

To take advantage of all the available multiple paths between R_0 and R_1 in ECMP routing in the same way as MPLS networks, we adjust the metric of $R_0 - R_2$ link to half of the default link metric. Therefore, three multiple paths available in ECMP routing are same as explicit multiple LSPs in the MPLS network. In this network configuration, Fig. 11 shows TCP performance by various types of senders and receivers. In Fig. 11-(a), we can see that all the proposed TCP modification schemes outperform the standard TCP method and the two explicit LSPs in the MPLS network. However, three explicit LSPs in the MPLS network utilize the maximum TCP throughput of the available bandwidth along multiple paths between R_0 and R_1. When the link loss rate of 0.001 is added (Fig. 11-(b)), fast retransmission based TCP modification methods still enhance TCP performance more than modified delayed ACK. Yet, flow-level multipath packet forwarding with three explicit multiple paths in the MPLS network achieves the maximum TCP throughput.

D. *Effects of Different Background Traffic*

In order to investigate the effect of background traffic on TCP performance, our simulation is performed on the very high performance Backbone Network Service (vBNS)-like topology (Fig. 12) [24]. The backbone links are configured as OC-12 (622 Mbps) and link delays are set in proportion to the distance between cities, while access links have 100 Mbps of bandwidth with 1 ms of delay. One hundred FTP connections (ten FTP

RED HAT 1012
RED HAT V. CAS

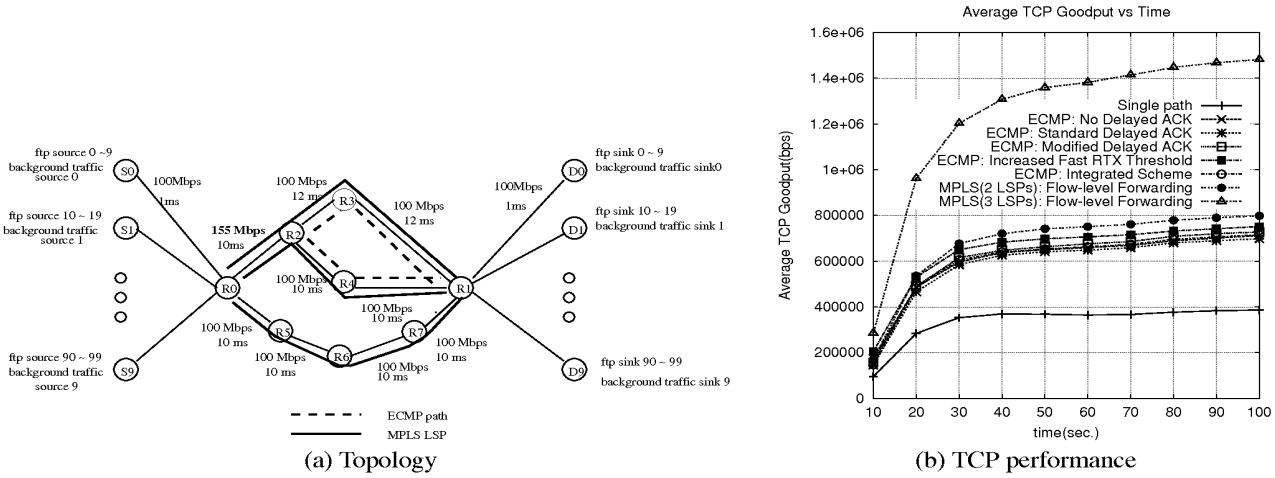


Fig. 10. TCP performance by explicit multiple paths in an MPLS network vs ECMP-based network.

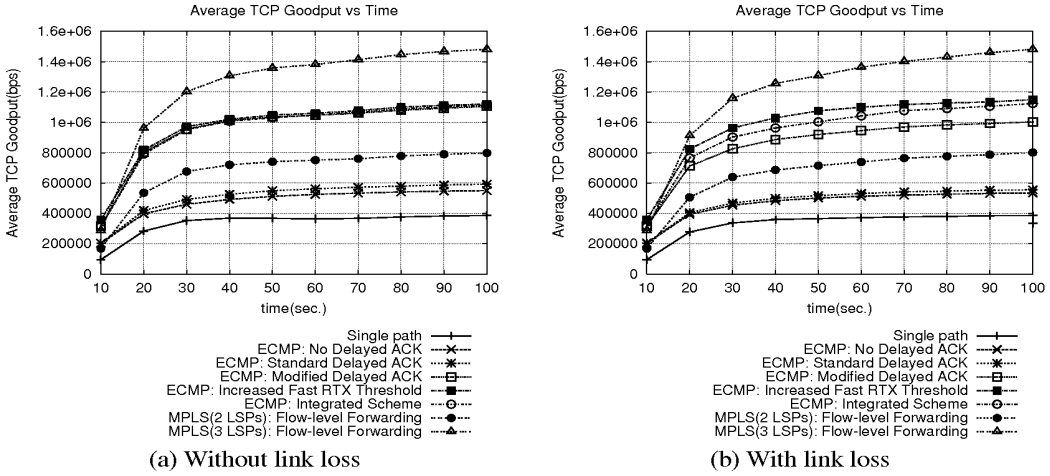


Fig. 11. TCP performance by explicit multiple paths in an MPLS network vs ECMP-based network after the link ($R_0 - R_2$) metric is adjusted.

senders/receivers per each leaf node) are concurrently tested along the path between San Francisco and Washington D.C with the Pareto-distributed background traffic (one Pareto source/sink per each leaf node). We changed the peak rate of each Pareto source for the different traffic load; lightly-loaded (30 Mbps), highly loaded (60 Mbps), and congested (70 Mbps).

In Fig. 13, it is shown that TCP performance over multiple paths is improved by the fast retransmission threshold heuristic, the modified delayed ACK mechanism, and the flow-level forwarding MPLS network under different background traffic loads. With proposed delayed ACKs, TCP performance increases by 43, 82, and 94% for the lightly-loaded, heavily-loaded, and congested path, compared with the single path case. Also, the adjusted fast retransmission threshold improves TCP goodput by 56, 82, and 92 %. When the integrated scheme is applied, TCP performance increases are 57, 84, and 95 %. With flow-level forwarding in MPLS networks, TCP goodput is increased by 61, 111, and 122 %. However, when the traditional TCP version of non-delayed or standard delayed ACKs is used in the ECMP routing, TCP performance is degraded for various background traffic loads, even lower than that of a single path in the lightly loaded environment. Although the path is lightly loaded, many out-of-order packet arrivals, which are considered as losses, still occur, and they reduce the sender's window by the fast retransmission/fast recovery.

On the other hand, if the short-lived TCP flows, found in WWW applications, use multipath routing, TCP performance enhancement is not so significant as in the case of ftp. However, as shown in the simulation results, the short-lived flows still benefit from multipath routing. In addition, if we consider that web sessions now include more long-lived and heavy file retrievals for multimedia applications, multipath routing will be-

RED HAT 1012
RED HAT V. CAS

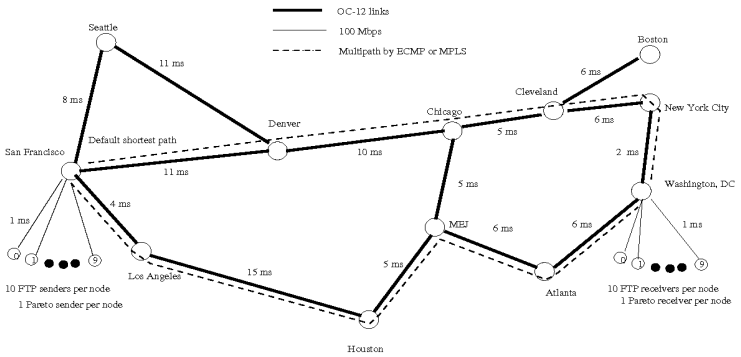
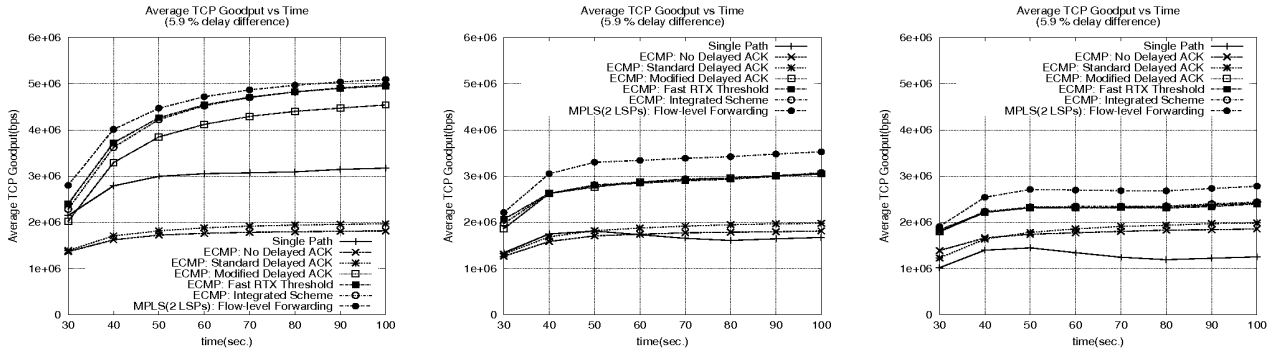


Fig. 12. vBNS-like network topology.



(a) Lightly loaded (b) Heavily loaded (c) Congested

Fig. 13. TCP performance by different background traffic.

come more useful.

V. CONCLUSION

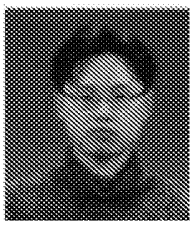
In this paper, we propose TCP modifications for multipath forwarding networks, and show that TCP performance is greatly improved even in a simple packet-level multipath forwarding networks. Increasing the fast retransmission threshold makes TCP senders wait for more than triple duplicate ACKs, which reduces the number of the fast retransmission and the fast recovery events. This method is suitable when a few TCP senders push long-lived flows to many receivers. The proposed delayed ACK method enhances TCP performance by generating less ACKs at TCP receivers, because out-of-order packet arrivals are recovered while ACKs are being delayed. This method achieves performance improvement similar to the sender-side solution, and is useful when sender modification is impossible. The integrated scheme, which both senders and receivers use TCP modification, also improves TCP performance. By the simulation results, it is seen that when the TCP modifications are employed, most out-of-order packet arrivals from different path delays are recovered and TCP performance is enhanced nearly by double, especially for two paths. Therefore, TCP modification methods will be practically useful for the conventional packet-level multipath network. On the other hand, it is verified that hashing-based flow-level multipath forwarding routers increase TCP performance in proportion to the number of paths in spite of the differences in the path delays. Moreover, it has been shown that explicit multiple paths in the MPLS network, when combined with flow-level multipath forwarding, improves both the end-to-end throughput and the network utilization by using the non-shortest paths. This feature can be used for the efficient Internet traffic engineering.

REFERENCES

[1] Y. Rekhter and T. Li, "A border gateway protocol," IETF RFC 1771, 1995.
[2] J. Moy, "OSPF version 2," IETF RFC 2328, 1998.
[3] R. Callon, "Use of OSI IS-IS for routing in TCP/IP and dual environments," IETF RFC 1195, 1990.
[4] D. Thaler and C. Hopps, "Multipath issues in unicast and multicast nexthop selection," IETF RFC 2991, 2000.
[5] J. Chen, P. Druschel, and D. Subramanian, "An efficient multipath forwarding method," in *Proc. IEEE INFOCOM'98*, 1998, pp. 1418–1425.
[6] S. Vutukury and J. J. Garcia-Luna-Aceves, "MDVA: A distance-vector multipath routing protocol, " in *Proc. IEEE INFOCOM'2001*, 2001, pp. 557–564.
[7] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," IETF RFC 3031, 2001.
[8] P. Newman, T. Lyon, and G. Minshall, "Flow labeled IP: A connectionless approach to ATM," in *Proc. IEEE INFOCOM'96*, 1996, pp. 1251–1260.
[9] Cisco NetFlow, Available at http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct.
[10] Z. Cao, Z. Wang, and E. Zegura, "Performance of hashing-based schemes

RED HAT 1012
RED HAT V. CAS

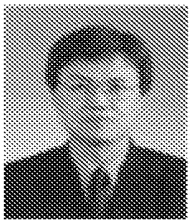
for internet load balancing," in *Proc. IEEE INFOCOM'2000*, 2000, pp. 332–341.
[11] J. Chen, "Multipath routing for large-scale networks," Ph.D. Dissertation, Rice University, 1999.
[12] R. Braden, "Requirements for internet hosts – communication layers," IETF RFC 1122, 1989.
[13] M. Allman, "On the generation and use of TCP acknowledgements," *ACM Computer Commun. Review*, vol. 28, no. 5, pp. 4–21, Oct. 1998.
[14] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, "The effects of asymmetry on TCP performance," in *Proc. ACM/IEEE MobiCom'97*, Sept. 1997, pp. 77–89.
[15] J. Padhye *et al*., "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM'98*, 1998, pp. 303–314.
[16] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 885–896, Dec. 1999.
[17] S. Bahk, and M. Zarki, "Dynamic multi-path routing and how it compares with other dynamic routing algorithms for high speed wide area networks," *ACM Computer Commun. Review*, vol. 22, no. 4, pp. 54–64, Oct. 1992.
[18] N. S. V. Rao and S. G. Batsell, "QoS routing via multiple paths using bandwidth reservation," in *Proc. IEEE INFOCOM'98*, 1998, pp. 11–18.
[19] A. Shaikh, J. Rexford, and K. G. Shin, "Load-sensitive routing of long-lived IP flows," in *Proc. ACM SIGCOMM'99*, 1999, pp. 215–226.
[20] P. Bhaniramka, W. Sun, and R. Jain, "Quality of service using traffic engineering over MPLS: An analysis," in *Proc. IEEE LCN'2000*, 2000, pp. 238–241.
[21] H. Saito, Y. Miyao, and M. Yoshida, "Traffic engineering using multiple multipoint-to-point LSPs," in *Proc. IEEE INFOCOM'2000*, 2000, pp. 894–901.
[22] S. Johnson, "Increasing TCP throughput by using an extended acknowledgement interval," Master's Thesis, Ohio University, June 1995.
[23] Network simulator - ns(version 2), Available at http://www-mash.cs.berkeley.edu/ns.
[24] very high performance Backbone Network Service (vBNS), Available at http://www.vbns.net.



Youngseok Lee received B.S and M.S. degrees in computer engineering from the Seoul National University, Seoul, Korea, in 1995 and 1997, respectively. Currently, he is studying toward the Ph. D degree at the School of Computer Science and Engineering, Seoul National University. He is mainly engaged in research on Internet traffic engineering and network planning.



Ilkyu Park received received B.S and M.S. degrees in computer engineering from the Seoul National University, Seoul, Korea, in 1999 and 2001, respectively. In 2001, he joined as a researcher the Distributed Virtual Reality Research Team, Electronics and Telecommunications Research Institute (ETRI), Taejon, Korea. His research interests include massive networked games, distributed system and high-speed network.



Yanghee Choi received B.S. in electronics engineering from Seoul National University, M.S. in electrical engineering from Korea Advanced Institute of Science, and Doctor of Engineering in Computer Science from Ecole Nationale Superieure des Telecommunications (ENST) in Paris, in 1975, 1977, and 1984 respectively. Before joining the School of Computer Engineering, Seoul National University in 1991, he has been with Electronics and Telecommunications Research Institute (ETRI) during 1977-1991, where he served as director of Data Communication Section, and Protocol Engineering Center. He was research student at Centre National d'Etude des Telecommunications (CNET), Issy-les-Moulineaux, during 1981-1984. He was also Visiting Scientist to IBM T. J. Watson Research Center for the year 1988-1989. He is now leading the Multimedia Communications Laboratory in Seoul National University. He is also director of Computer Network Research Center in Research Institute of Advanced Computer Technology (RIACT). He was editor-in-chief of Korea Information Science Society journals. He was chairman of the Special Interest Group on Information Networking. He has been associate dean of research affairs at Seoul National University. He is now president of Open Systems and Internet Association of Korea. His research interest lies in the field of multimedia systems and high-speed networking.

RED HAT 1012
RED HAT V. CAS

Transport Level Mechanisms for Bandwidth Aggregation on Mobile Hosts

Luiz Magalhaes and Robin Kravets
Department of Computer Science
University of Illinois, Urbana-Champaign
1304 W Springfield Avenue
Urbana, IL 61801
{magalhae,rhk}@cs.uiuc.edu

Abstract

Present mobile computing does not support the simultaneous use of multiple heterogeneous network interfaces for a single transport layer connection. In this paper, we describe a solution for channel aggregation at the transport layer, which provides increased bandwidth to mobile nodes. We present R-MTP (Reliable Multiplexing Transport Protocol), a rate-based reliable transport protocol capable of multiplexing data from a single application data stream across multiple network interfaces. Due to the lossy nature of wireless links in mobile environments, R-MTP tracks packet interarrival time for discrimination between congestion-based and transmission-based losses as well as better bandwidth estimation. The challenges to such a reliable protocol lie in the coordination of packets across streams with varying channel characteristics. Our experimental results validate R-MTP's bandwidth estimation and loss characterization techniques. Successful bandwidth aggregation is demonstrated in ideal and lossy environments

1 Introduction

With the growing popularity of wireless access, coverage areas are growing and often overlapping, enabling more than one communication technology in a specific area. This fact combined with the availability of hosts with multiple interfaces enables the simultaneous use of multiple communication channels, adding an untapped source of additional bandwidth to mobile networking systems. Current channel aggregation techniques are limited to identical channels across a homogeneous link layer. The goal of our research is the aggregation of available resources from multiple heterogeneous channels to create a virtual end-to-end channel with better characteristics than each channel alone. To this end, we have designed and developed R-MTP (Reliable Multiplexing Transport Protocol), a rate-based transmission protocol for the reliable transmission of bulk data to mobile nodes. The use of R-MTP preserves end-to-end semantics, transparently providing the application with the simultaneous use of multiple channels by multiplexing data from the application across a set of available channels. Our approach exposes link-layer connectivity and per channel resource information to the transport layer, allowing R-MTP to adapt to both changes in available bandwidth on each channel and changes in availability of channels. As the available channel resources change, R-MTP adapts, changing the fraction of flow that is being sent on each channel and adding or removing channels as necessary. For each channel, R-MTP monitors delay and interarrival time for use with bandwidth estimation and congestion control. In addition, R-MTP uses information generated by its rate-based mechanism (interarrival time) for the classification of losses (i.e. congestion losses vs. transmission errors) on the wireless link.

The key contribution of our research is the aggregation of resources across multiple heterogeneous channels in a mobile environment. The simultaneous use of communication channels from multiple technologies enhances communication support for mobile devices in several ways. The first advantage is the increase in bandwidth through bandwidth aggregation across channels. In traditional wired communication, the bottleneck link to a host is often some router in the network. Additional bandwidth at the endhost will not alleviate this problem. In wireless environments, the bottleneck link is often the last hop. Combining multiple channels into a single virtual channel can increase the capacity of the last hop. The second advantage is to free the system from committing to a single choice of a link-layer channel. By providing information about currently available channels, the system can choose some subset of these channels for its communication. In addition, in the event of a blackout or congestion in any of the active link layers, the use of multiple link layers will enable the host to stay connected. Using every available link layer all the time may not always be the best solution. The addition of a resource poor channel may actually hurt performance, the complexities of which we discuss later in the paper. In

1092-1658/01 $17.00 © 2001 IEEE

RED HAT 1012
RED HAT V. CAS

general, the choice of which channels to use depends on the constraints set by the user (e.g. cost), the protocol (e.g. performance) and the operating system (e.g. power consumption).

The remainder of this paper is organized into four sections. The next section contains related research and design decisions. R-MTP is described in Section 3, where we present our channel abstraction and the mechanisms for bandwidth estimation, reliability, flow control and channel management. The experimental results are discussed in Section 4, and Section 0 contains the conclusions and future research directions.

2 Communication Channel Multiplexing

The aggregation of multiple communication channels has been attempted at different layers of the protocol stack. The novel aspect our multiplexing solution is that we expose multiplexing to the transport level. The base-stations or access points do not need to be modified because they are not aware that they are carrying flows that have been aggregated into one virtual channel. The hosts at the end-points are aware of the multiplexing, thereby providing an end-to-end solution and allowing the use of heterogeneous technologies with diverse attachment points. In contrast, the current solutions work at the link layer.

At the link layer, such aggregation for serial devices has been implemented in Linux, Windows and many commercial routers. For these solutions the characteristics of all communication channels must be identical. Linux, Cisco IOS and Sun also allow for the use of multiple Ethernet adaptors in the same fashion, in what is called respectively "bonding", "etherchannel" and "trunking". This type of channel aggregation must be done between the same two endpoints because this is a link layer solution, and transparent to the upper level protocols. ATM attempts to provide a variety of rates and flexibility in allocating bandwidth for commercial services through channel aggregation. The aggregation of many lower bandwidth channels into a larger pipe is called "reverse multiplexing" [1], and is now part of the ATM specification. Since ATM depends on in-order delivery for cell reassembly, much effort has been put into maintaining the strict synchronization of different flows. Similar work has also been done in the aggregation of bandwidth [2] in wireless links using the facilities of PPP (multilink) [3]. The mechanisms described in this paper are more general, allowing the use of any interface that supports IP.

The addition of multiplexing at the transport layer introduces an increase in the occurrence of out-of-order delivery of packets due to the different delays experienced on the different channels. Large differences in delay have two effects on multiplexing transport protocols. First, a packet being transmitted over a channel with long delay may appear to be lost if the loss detection methods do not consider the channel being used and the characteristics of that channel. Gap detection can be used on a per channel basis, but has no meaning across channels. Similarly, the use of duplicate acknowledgements for quick notification of lost messages is meaningless if the sending channel is not taken into account. Second, large differences in delays across channels will affect the performance of the protocol. Application-level framing [4] may free protocols from providing resequencing and allow applications to deal with out-of-order reception, enabling the faster use of incomplete data; however, most applications currently in use require that transport protocols provide in-order delivery.

3 R-MTP

R-MTP is a protocol designed for the reliable transmission of bulk data to mobile systems that have access to multiple link-layer technologies. R-MTP is designed as a multiple channel, rate-based protocol that uses selective acknowledgements[5] for reliability, and bandwidth estimation for flow and congestion control. Rate-based protocols have a long history ([6],[7],[8]). In this section, we present R-MTP's architecture and discuss the parameters used per channel and across multiple channels. We then discuss bandwidth estimation, reliability and flow control. Due to space constraints, we omit the mechanism for adding and removing channels, and the format of the packet headers (See [9]).

3.1 Protocol Architecture and Parameters

Our approach defines the architecture of an end-to-end transport layer that supports the aggregation of resources from multiple end-to-end network layer channels. The transport layer provides a multiplexing service that maps a single virtual application layer channel across these multiple network layer channels. The application submits data units to the end-to-end transport layer for transmission; the transport layer processes them and interacts with the various network channels to determine which channel to use for transmission. The transport layer on the receiving side processes the incoming data units from all channels and presents them to the application layer in a meaningful way, where meaningful is defined by the specific application requirements.

RED HAT 1012
RED HAT V. CAS

Our goal is to design the transport layer to adapt its behavior to changes in the availability of individual link-layer channels, as well as the characteristics of the specific channels to which the transport layer has access. On startup, a collection of one-way network channels is established and each of these channels is probed via the packet pair method [10] to determine available bandwidth on the channel. This technique provides the current value of the smallest interval at which packets can be sent without experiencing queueing delays. Delay, or round trip time, is measured by recording the time a frame was sent and its acknowledgment received. These measurements define the main parameters of R-MTP, which are used by its reliability, congestion control and flow control mechanisms.

For reliability and sequencing, R-MTP uses a window-based reliability mechanism with selective acknowledgements. The size of the reliability window, *window size*, defines how much space is available at the receiver for active messages, where an active message is a message that may still be accepted and buffered by the receiver. The value of *window size* depends on the bandwidth and delay estimates for all channels currently in use. Since a fixed size bitmap is used to implement the selective acknowledgements, the window size is fixed during the lifetime of the connection to maintain a constant header size. Processing is simplified if the header is word aligned, therefore, *window size* is defined as the minimum multiple of 32 that is greater than twice the sum of the bandwidth delay product of each channel. The bandwidth delay product is the number of packets in flight at each time. The reliability window must accommodate enough packets so protocol processing does not stop if a packet is lost. This requires the window to be large enough to accommodate all the packets transmitted from the time a packet was lost until a replacement arrives.

R-MTP tracks the available buffer space at the receiver with the variable *free buffer space*. Similarly to TCP, the receiver maintains a receive buffer, equal in size to the reliability window, for buffering messages that have already been processed by R-MTP, but not yet passed up to the application. The amount of free space in this buffer dictates how far the sender can slide its send window upon receipt of acknowledgements from the receiver. If there are no more free buffers, R-MTP can continue processing messages in its reliability window, but will not be able to advance its reliability window. R-MTP uses a field in its header to report the value of *free buffer space* to the other side of the protocol and maintains the parameter *maximum receiver rate* to track the maximum rate at which a receiver can process frames. If transmission is limited by the processing power of a receiver, R-MTP caps its transmission rate at *maximum receiver rate*.

R-MTP tracks the maximum bandwidth per channel both to avoid trying to send too much data across a channel and to make sure a channel is used to its capacity. Since R-MTP is a rate-based protocol, R-MTP translates this information into a *maximum rate* that is sustainable on the channel. To use all available bandwidth, R-MTP continuously probes active communication channels. Probing may cause packet loss if it is done when a channel is operating at its limit, since probing beyond the maximum available bandwidth will momentarily exceed that maximum. If the maximum channel bandwidth is known, and the *maximum rate* is set to that value, R-MTP will stop probing when the maximum bandwidth of a channel is achieved.

The ideal rate is an elusive target. It should be as close as the maximum allowed by the channel before causing congestion, but since channels are shared, the ideal rate will change over time. The interarrival time measured at startup is used to calculate the *initial rate*. The ideal rate is the inverse of the minimum period that the channel can support without causing congestion, which is given by the interarrival time measured by the packet pair method. Since this measurement may contain errors, the *initial rate* is defined as half the calculated rate. The rate will track the available bandwidth, and eventually converge to the inverse of the interarrival time or the *maximum rate*.

3.2 Bandwidth estimation

Bandwidth estimation is at the heart of the protocol. It sets the rate at which each channel sends frames, defines how information is multiplexed on the available channels, and is the major input to infer if losses are congestion related or caused by the medium ([11],[12],[13]). The protocol keeps track of the *interarrival times* and *jitter* for each frame received, and a cumulative *long run jitter*. The *interarrival time* of a frame is the difference in the arrival time of the previous frame and the current frame in this channel. If a frame is lost then it has no meaning, and it is not calculated. The *jitter* is the difference between the measured *interarrival time* and the *rate* of the channel, which is calculated at the receiver and sent in the frame header. The *long run jitter* should hover around zero, because the value of the jitter may be positive or negative, when delays in the previous frame may case the current frame to appear to arrive early (Figure 1). If the *long run jitter* starts to grow, the system is experiencing congestion, since each individual jitter is getting larger without a negative jitter to cancel out the increase.

RED HAT 1012
RED HAT V. CAS

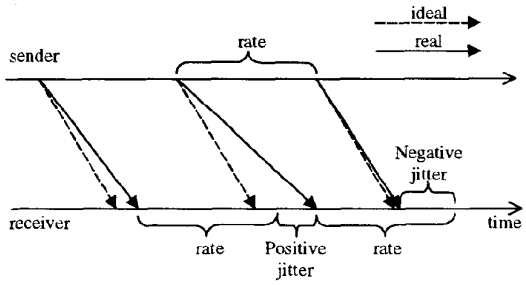


Figure 1: Negative and positive jitter values caused by the variance in propagation delays

By tracking the values of *jitter* and *long run jitter*, R-MTP infers if losses are due to congestion or due to the medium. If *long run jitter* is positive and the last *jitter* value was large when a loss was detected, then the loss is deemed a congestion loss and the protocol slows down its sending rate by changing its rate on the receiver and sending the new rate to the sender. Otherwise, the loss is considered a medium loss, and the rate is unchanged.

Using a congestion avoidance mechanism similar to WTCP's [7], R-MTP does not depend solely on packet loss to sense congestion. If two consecutive values of the *jitter* show a growing trend and *long run jitter* is positive, the protocol reacts to the incipient congestion. The growing jitters combined with a net increase in the *long run jitter* may mean that somewhere along the path a router queue is growing, increasing packet latency. In this case, the protocol slows down to avoid congestion.

Measuring *interarrival times* and *jitter* allows the protocol to sense bandwidth scarcity. It does not allow the protocol to detect bandwidth abundance. If more than enough bandwidth is available to support a certain *sending rate*, the *interarrival times* will converge to the *sending rate*, but will never surpass it. To use all available bandwidth the protocol must probe the channels periodically and adjust the rate accordingly. The method used to probe available bandwidth is the packet pair method. Two packets are sent back-to-back to probe the channel and measure the current *minimum interarrival time*. If the *minimum interarrival time* is smaller than the current period (inverse of the current *rate*), the current *rate* is increased by a fraction of the difference. If the available bandwidth is a constant, then the *rate* will converge to the optimal rate.

In general, the available bandwidth is a dynamic value, changing according to the channel usage. Measurements are made at the receiver, and fed back to the sender. Thus, the protocol will always trail changes. To minimize time under non-optimal channel usage, it would be good to probe often. Since probing disrupts channel regularity, over which congestion estimation is made, the maximum *probing rate* is once every four packets. The *probing rate* is adjusted to match channel conditions. When the protocol drops its rate because of congestion, the *probing rate* also drops. If the channel rate reaches the *maximum rate* defined for the channel, probing is disabled.

3.3 Reliability

R-MTP uses retransmission-based reliability and gap-detection for identifying losses. The sender is notified that frames have arrived at the receiver by receiving acknowledgements. There are two types of acknowledgments: cumulative and selective. The cumulative acknowledgment carries the number of the next expected frame (last received + 1). The selective acknowledgment is a bit map of the state of the received queue, with bit 0 being the position of the next expected frame. The sender keeps an individual bit map for each channel to record which frames were sent on the channel.

Gap detection works by checking the selective acknowledgement for gaps in the sequence of each channel. Every time a frame is sent through a channel, it is marked in that channel's bitmap. When the selective acknowledgement is received, it is compared with each channel bitmap. If gaps are found, the missing frames are retransmitted. Once a frame is retransmitted, the protocol can no longer rely on sequencing to detect gaps, since the retransmitted frame is out of order. Two additional global bit maps are kept to track information about retransmissions. One bitmap, *retransmitted frames*, tracks all retransmitted frames to prevent multiple retransmissions of the same packet before it has had a chance to be successfully received and acknowledged. In order to determine when a retransmission has been lost, R-MTP maintains a second bit map for *marker frames* that is used to indicate the frame that was sent immediately after a retransmission. If an acknowledgement for the *marker frame* arrives before the acknowledgement for the retransmitted frame, the retransmitted frame is deemed lost and sent again.

Marker frames are needed because even after a gap is found and a lost frame retransmitted, all acknowledgements generated before the lost frame arrives at the receiver will contain the same gap. If a frame were retransmitted every time a gap was detected, there would be many duplicated retransmissions. To prevent unnecessary retransmissions, a frame in the *retransmitted frames* bitmap can only be retransmitted again after its

RED HAT 1012
RED HAT V. CAS

marker frame in the *marker frames* bitmap has been cleared. The position in the *marker frames* bitmap is cleared when an acknowledgement or a gap for that position is found. This allows the protocol to do all its retransmissions without using explicit timers.

The reliability window must buffer enough frames to allow the acknowledgement for a frame to be received in time to prevent its retransmission. If more than one channel is being used, then the Minimum Window Size (MWS) is given by a ratio of the rates and the longest propagation delay. The MWS is defined as the number of buffers needed to accommodate all packets that are transmitted between the time a packet is sent on the channel with longest propagation delay and the time that its acknowledgement is received. The total propagation time is the one-way propagation delay of the slowest channel (packet) plus the one-way propagation delay of the fastest channel (ack). The number of packets transmitted in this interval in a channel is the total propagation time times the rate of the channel. The total of packets is the sum of the number of packets transmitted in each channel. Therefore, MWS is given by the expression below:

$$MWS = \sum_{i} (delay\_slowest\_channel + delay\_fastest\_channel) * rate\_channel(i)$$

3.4 Flow Control

There are two mechanisms used for flow control. One is the *maximum rate* negotiated on startup. The *maximum rate* defines a hard limit on the number of packets sent per second – even if the protocol measures that the available bandwidth is greater than it is currently using, it will not surpass the *maximum rate*. When the protocol reaches the *maximum rate* it will stop probing until the rate drops. The second mechanism is the receiver queue. This is an ancillary queue that smoothes the bunching effect that the mismatch of rates and delays generates, and is implemented as a part of the receiver window. The flow control works in the same way as TCP, but with a fixed size queue: the packet header carries how many slots are left on the receiver queue. The sender will stop sending data frames if the queue gets full, halve the rate and send only acknowledgements until it gets acks from the receiver that shows there is space again in the receiver queue, at which time normal processing resumes. The flow control and reliability queues are two separate queues, the data in the flow control queue having already been acknowledged.

4 Experimental Results

R-MTP provides a reliable transport service using bandwidth aggregation and is optimized for wireless environments. The experiments in this section are designed to demonstrate the effectiveness of R-MTP as a reliable transport protocol, and R-MTP's bandwidth aggregation. We describe our test setup, and the experiments conducted. Due to space constraints, we selected two sets of experiments. In the first set, we address issues related to the performance of R-MTP in lossless environments. The experiments compare the performance of R-MTP to the performance of TCP over various link layer technologies. The second set demonstrates the effectiveness of R-MTP's bandwidth aggregation techniques, both in lossless and lossy environments. Other experiments (found in [9]) attest that R-MTP is TCP-friendly, and analyze in greater depth its reaction to loss and congestion.

The experiments use a user-space implementation of R-MTP that encapsulates its data in UDP packets. UDP already offers a *port number* and *checksum* so these variables were omitted from R-MTP's header. For each run, the test program sends one thousand packets of 1400 bytes, using both R-MTP and TCP. The receiving program records the arrival time of each packet, and the graphs plot the cumulative time of arrival of each packet and their interarrival time. The tests cover a sample of wireless link layer technologies: three types of infrared devices and two types of wireless Ethernet. Infrared has two different standards for point-to-point communication: IRLan, a LAN emulation over infrared which is an IRDA standard, and IRNet.

4.1 Performance of a single link layer under lossless conditions

Table 1 presents a comparison of R-MTP and TCP over wireless channels with no external losses. In general, we expected TCP to perform better than R-MTP since R-MTP is implemented in user space. Surprisingly, R-MTP performs better than TCP even when no external losses are introduced. On slower network interfaces, a node's CPU can feed data to the network interface faster than the data can be transmitted on the link. This allows a single transport layer connection to use all available bandwidth of this link. Under this condition, TCP's bandwidth probing can cause losses by congesting the link. This results in a net performance below R-MTP's, whose

RED HAT 1012
RED HAT V. CAS

bandwidth probing rarely causes losses in these experiments.

Link Layer Technology	Total Run Time R-MTP (sec)	Total Run Time TCP (sec)
IRLan	159.5 ± 3	160.9 ±2
IRNet	156.0 ± 4	160.0 ± 3
Wavelan	10.69 ± 0.7	9.89 ± 0.4
Aviator	8.64 ± 0.5	9.03 ± 0.6

Table 1: Comparison of R-MTP and TCP over various wireless technologies

4.2 Bandwidth Aggregation

Dealing with losses provides a good argument for using multiple link layers simultaneously, even when bandwidth aggregation does not seem compelling. Bandwidth aggregation may significantly increase the total bandwidth available to a mobile host if the link layers have similar capacities; (i.e. aggregating two 115Kbps channels is a good choice). On the other hand, it is not intuitive that aggregating a 1.2 Mbps and an 115Kbps channel will add enough bandwidth to offset the cost of multiplexing. The result of both bandwidth aggregations can be seen in Figure 2 and Figure 3 below. Figure 2 shows the aggregation of two infrared channels at 115Kbps, an IRLAN channel and an IRNET channel. The aggregate channel is considerably faster than each individual channel. This is not the case for the example shown in Figure 3, which shows that the channel resulting from the aggregation of an Aviator wireless Ethernet and an infrared IRLAN connection is only slightly faster than the individual Ethernet channel. In fact, the multiplexing overhead makes the composite channel slower at some points.

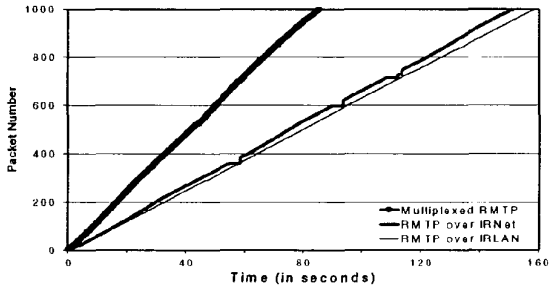


Figure 2: Bandwidth aggregation on IRNet and IRLAN

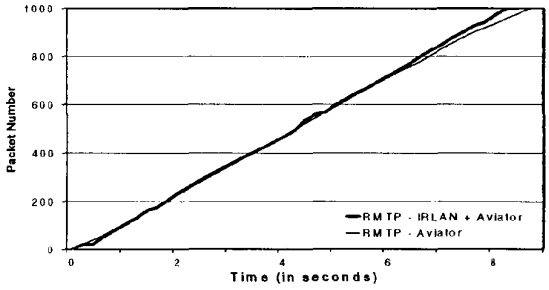


Figure 3: Bandwidth aggregation on Aviator and IRLAN

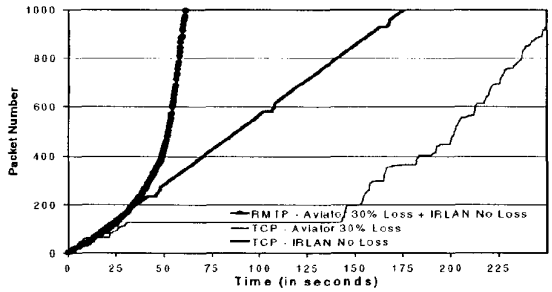


Figure 4: Comparison of TCP over Aviator with 30% loss and over IRLAN with no loss to R-MTP multiplexed over the same two channels

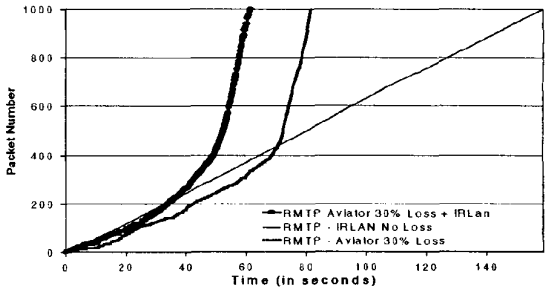


Figure 5: Comparison of R-MTP over Aviator with 30% loss and over IRLAN with no loss to R-MTP multiplexed over the same two channels

542
RED HAT 1012
RED HAT V. CAS

Bandwidth aggregation with such mismatching channel characteristics can be justified in certain scenarios. Figure 4 shows an experiment with the same infrared interface and wireless Ethernet of Figure 3, but with a 30% loss rate on the Ethernet. A TCP run, which normally takes 10 seconds to complete on the wireless Ethernet link, now takes 25 times longer. Now the IRLAN link, lossless in this example, is performing better than the Ethernet link, and completes the run in 160 seconds. Figure 5 shows R-MTP using Ethernet, IRLAN and both multiplexed. While R-MTP has better behavior on the lossy link, finishing the run on the wireless Ethernet alone in 81.5 seconds, adding the much lower bandwidth link actually improves the performance by 25% in this case (61.2 seconds).

Beyond the significant improvement in performance by using two seemingly mismatched interfaces, these results make a case for using all available interfaces all the time. While this must be offset by power and cost considerations, because it is impossible to measure a priory if an interface is lossy, or to predict noise, using all communication resources that are available creates a virtual channel that is more resilient than any channel alone.

5 Conclusion and Future Work

Mobile nodes using wireless communication operate in less than ideal environments. Low bandwidth links limit application throughput, lossy channels challenge transport protocols optimized for congestion-based losses, and constantly changing channel availability causes breaks in transmission. The solutions presented in this paper address these problems through the use of end-to-end channel aggregation. With such techniques, additional bandwidth can be added to the mobile host's bottleneck link. The use of multiple channels can also alleviate the effect of losses or complete outages on one particular channel.

Our experiments demonstrate the performance of R-MTP in various wireless environments. The success of bandwidth aggregation across similar links supports our intuition that adding bandwidth to the bottleneck link increases throughput for the application. Most importantly, we show that aggregation across two channels with an order of magnitude difference in capacity is beneficial if the high bandwidth channel is experiencing significant loss.

Two very important issues are part of our future work. We understand the use of multiple interfaces will have a large impact on the energy consumption of a mobile node, and we are currently investigating the effect of using R-MTP on energy consumption. Our intuition is that the use of multiple interfaces can minimize transfer time and so conserve the total energy used by a mobile (system+network) in some scenarios. The second is related to mobility. Additional testing must be done to gauge R-MTP's success in mobile environments. Comparison with a solution using Mobile IP and TCP, and the changes proposed in TCP to make it mobility-aware [14], can demonstrate the effect of R-MTP's transport layer mobility solution.

6 REFERENCES

1. Chiussi, F.M., D.A. Khotimsky, and S. Krishnan. *Generalized inverse multiplexing of switched ATM connections.* in *Proceedings of the IEEE Conference on Global Communications (GlobeCom '98).* 1998.
2. Snoeren, A.C. *Inverse Multiplexing for Wide-Area Wireless Networks.* in *Proceedings of the IEEE Conference on Global Communications (GlobeCom '99), Global Internet Symposium.* 1999.
3. Sklower, K., et al., *The PPP Multilink Protocol, RFC1990.* 1996.
4. Clark, D.D. and D.L. Tennenhouse, *Architectural Considerations for a New Generation of Protocols,* in *Proceedings of the SIGCOMM '90 Symposium.* 1990. p. 200-208.
5. Mathis, M., et al., *TCP Selective Acknowledgement Options.* 1996, Internet Engineering Task Force.
6. Clark, D.D., M. Lambert, and L. Zhang, *NETBLT: A High Throughput Transport Protocol.* 1988.
7. Sinha, P., et al. *WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks.* in *ACM Mobicom '99.* 1999.
8. Rejaie, R., M. Handley, and D. Estrin. *RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet.* in *IEEE Infocom 99.* 1999.
9. Magalhaes, L., *R-MTP: Reliable Multiplexing Transport Protocol.* 2001, University of Illinois at Urbana-Champaign.
10. Keshav, S. *A Control-Theoretic Approach to Flow Control.* in *Proceedings of the SIGCOMM '92 Symposium.* 1992.
11. Balakrishnan, H., et al., *Improving TCP/IP Performance over WIreless Networks,* in *First ACM International Conference on Mobile Computing and Networking (MOBICOM).* 1995.
12. Biaz, S. and N. Vaidya, *Discriminating congestion losses from wireless losses using inter-arrival times at the receiver.* 1998, Technical Report 98-014, CS Dept., Texas A&M University.
13. Caceres, R. and L. Iftode, *Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments.* IEEE Journal on Selected Areas in Communications, 1995. **13**(5).
14. Snoeren, A. and H. Balakrishnan. *An End-to-End Approach to Host Mobility.* in *ACM Mobicom '99.* 2000.

RED HAT 1012
RED HAT V. CAS

Analysis and Comparison of TCP Reno and Vegas

Jeonghoon Mo, Richard J. La*, Venkat Anantharam, and Jean Walrand
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
{jhmo, hyongla, wlr, ananth}@eecs.berkeley.edu

July 13, 1998

Abstract

We propose some improvements of TCP Vegas and compare its performance characteristics in comparison with TCP Reno. We demonstrate through analysis that TCP Vegas, with its better bandwidth estimation scheme, uses the network resources more efficiently and is more fair than TCP Reno. Simulation results are given and support the results of the analysis.

1 Introduction

Following a series of congestion collapses starting in October of '86, Jacobson and Karels developed a congestion control mechanism, which was later named TCP Tahoe [9]. Since then, many modifications have been made to the transmission control protocol (TCP) and several different versions of TCP have been implemented [9, 10].

Recently Brakmo et al. [2] have proposed a new version of TCP, which is named TCP Vegas, with a fundamentally different congestion avoidance scheme from that of TCP Reno and claimed that TCP Vegas achieves 37 to 71 percent higher throughput than TCP Reno. Ahn et al. [1] have evaluated the performance of TCP Vegas on their wide area emulator and shown that TCP Vegas does achieve higher efficiency than TCP Reno and causes much less packet retransmissions. However, they have observed that TCP Vegas when competing with other TCP Reno connections, does not receive a fair share of bandwidth, i.e., TCP Reno connections receive about 50 percent higher bandwidth.

*address all correspondence to the second author

RED HAT 1012
RED HAT V. CAS

Floyd has observed that TCP Reno is biased against the connections with longer delays [5, 7] . The intuition behind this behavior is as follows. While a source does not detect any congestion, it continues to increase its window size by one during one round trip time. Obviously the connections with shorter delay can update their window sizes faster than those with longer delays, and thus steal higher bandwidths. Based on this observation, Floyd and Jacobson [6] have proposed a *constant rate adjustment* algorithm. Henderson et al. [8] have simulated a variation of this scheme and reported that if the rate of increase of the window size is not excessive, this scheme is not harmful to the other connections that use a different version of TCP. Moreover, as expected, this scheme results in better performance for the connections with longer delays. However, choosing the parameters for such algorithms is still an open problem.

In this paper, we explain the performance characteristics of TCP Vegas such as fairness and incompatibility with TCP Reno. We also discuss a few problems with the original TCP Vegas. The rest of the paper is organized as follows. In section 2 we describe the congestion detection and avoidance schemes of TCP Reno and TCP Vegas. Section 3 presents a few problems intrinsic in the original TCP Vegas and explains a possible solution to them. In section 4 we discuss the fairness of TCP Vegas using a simple closed fluid model. Section 5 compares TCP Vegas with TCP Reno and explains the incompatibility between them. We confirm our analysis with simulation results in section 6, and finish with some conclusion and future problems.

2 Congestion Avoidance Mechanism

In order to make an efficient use of network bandwidth, TCP controls its flow rate using the feedback from the network. In order to control its flow rate, TCP needs to estimate the available bandwidth in the network using a *bandwidth estimation scheme*. Among many new features implemented in TCP Vegas, the most important difference between TCP Vegas and TCP Reno lies in this bandwidth estimation scheme used to estimate the available bandwidth. As will be shown shortly, TCP Reno views packet losses as a signal of network congestion, while TCP Vegas uses the difference in the expected and actual rates to adjust its window size. This fundamental difference in the bandwidth estimation schemes enables TCP Vegas to utilize the available bandwidth more efficiently.

RED HAT 1012
RED HAT V. CAS

2.1 TCP Reno

TCP Reno induces packet losses to estimate the available bandwidth in the network. While there are no packet losses, TCP Reno continues to increase its window size by one during each round trip time. When it experiences a packet loss, it reduces its window size to one half of the current window size. This is called *additive increase and multiplicative decrease*. Additive increase and multiplicative decrease algorithm is based on the results given in [3]. They have shown that such an algorithm leads to a fair allocation of bandwidth. TCP Reno, however, fails to achieve such fairness because TCP is not a synchronized rate based control scheme, which is necessary for the convergence.

As can be easily seen, the congestion avoidance mechanism adopted by TCP Reno causes a periodic oscillation in the window size due to the constant update of the window size. This oscillation in the window size leads to an oscillation in the round trip delay of the packets. This oscillation results in larger delay jitter and an inefficient use of the available bandwidth due to many retransmissions of the same packets after packet drops occur.

The rate at which each connection updates its window size depends on the round trip delay of the connection. Hence, the connections with shorter delays can update their window sizes faster than other connections with longer delays, and thereby steal an unfair share of the bandwidth. As a result, TCP Reno exhibits an undesirable bias against the connections with longer delays [5, 7].

2.2 TCP Vegas

TCP Vegas adopts a more sophisticated bandwidth estimation scheme. It uses the difference between expected and actual flows rates to estimate the available bandwidth in the network. The idea is that when the network is not congested, the actual flow rate will be close to the expected flow rate. Otherwise, the actual flow rate will be smaller than the expected flow rate. TCP Vegas, using this difference in flow rates, estimates the congestion level in the network and updates the window size accordingly. Note that this difference in the flow rates can be easily translated into the difference between the window size and the number of acknowledged packets during the round trip time, using the equation

$$Diff = (Expected - Actual)\ BaseRTT, \tag{1}$$

where *Expected* is the expected rate, *Actual* is the actual rate, and $BaseRTT$ is the minimum round trip time. The details of the algorithm are as follows:

RED HAT 1012
RED HAT V. CAS

1. First, the source computes the expected flow rate $Expected = \frac{CWND}{BaseRTT}$, where $CWND$ is the current window size and $BaseRTT$ is the minimum round trip time.

2. Second, the source estimates the current flow rate by using the actual round trip time according to $Actual = \frac{CWND}{RTT}$, where RTT is the actual round trip time of a packet.

3. The source, using the expected and actual flow rates, computes the estimated backlog in the queue from *Diff = (Expected-Actual) BaseRTT*.

4. Based on *Diff*, the source updates its window size as follows.

$$CWND = \begin{cases} CWND + 1 & if\ Diff < \alpha \\ CWND - 1 & if\ Diff > \beta \\ CWND & otherwise \end{cases}$$

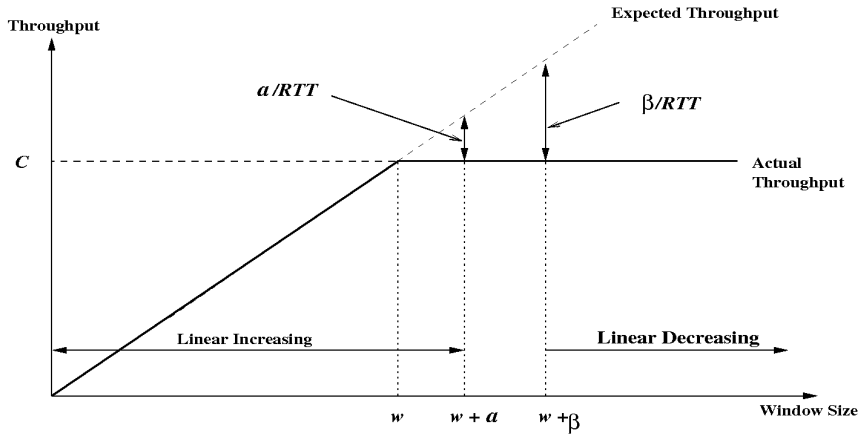


Figure 1: Window control of TCP Vegas

Figure 1 illustrates the behavior of TCP Vegas. Consider a simple network with one connection and one link with capacity C. Let $BaseRTT$ be the minimum round trip delay. The throughput of this connection is $\frac{window}{BaseRTT}$ when $window < C \times BaseRTT$. In the figure, w corresponds to the window size where $window = C \times BaseRTT$. When $window > w$, queue starts to build up and *(Expected - Actual)* > 0. TCP Vegas increases the window size by one during the next round trip time if $window < w + \alpha$ and decreases the window size by one if $window > w + \beta$. Otherwise, it leaves the window size unchanged. In the figure, *Diff* is user i's estimated backlogged queue size in the link. This will be explained in more details in section 3. TCP Vegas tries to keep at least α packets but no more than β packets in the queues. The reason behind this is that TCP

RED HAT 1012
RED HAT V. CAS

Vegas attempts to detect and utilize the extra bandwidth whenever it becomes available without congesting the network. Thus, when there is only one connection, the window size of TCP Vegas converges to a point that lies between $w+\alpha$ and $w+\beta$. Note that this mechanism is fundamentally different from that used by TCP Reno. TCP Reno always updates its window size to guarantee full utilization of available bandwidth, leading to constant packet losses, whereas TCP Vegas does not cause any oscillation in window size once it converges to an equilibrium point.

3 TCP Vegas

TCP Vegas has a few problems that have not been discussed before, which could have a serious impact on the performance. One of the problems is the issue of rerouting. Since TCP Vegas uses an estimate of the propagation delay, *baseRTT*, to adjust its window size, it is very important for a TCP Vegas connection to be able to have an accurate estimation. Rerouting a path may change the propagation delay of the connection, and this could result in a substantial decrease in throughput. Another important issue is the stability of TCP Vegas. Since each TCP Vegas connection attempts to keep a few packets in the network, when their estimation of the propagation delay is off, this could lead the connections to inadvertently keep many more packets in the network, causing a persistent congestion. La et el. [12] have investigated these issues. We summarize the results here. For more details, refer to [12].

3.1 Rerouting

TCP Vegas that was first suggested by Brakmo et el. [2] does not have any mechanism that handles the rerouting of connection. In this section, we will show that this could lead to strange behavior for TCP Vegas connections. Recall that in TCP Vegas *BaseRTT* is the smallest round trip delay, which is an estimate of the propagation delay of the path .

First, if the route of a connection is changed by a switch, then without an explicit signal from the switch, the end host cannot directly detect it. If the new route has a shorter propagation delay, this does not cause any serious problem for TCP Vegas because most likely some packets will experience shorter round trip delay and *BaseRTT* will be updated. On the other hand, if the new route for the connection has a longer propagation delay, the connection will not be able to tell whether the increase in the round trip delay is due to a congestion in the network or a change in the route. Without this knowledge the end host will interpret the increase in the round trip delay as a

RED HAT 1012
548
RED HAT V. CAS

sign of congestion in the network and decrease the window size. This is, however, the opposite of what the source should do. When the propagation delay of connection i is d_i, the expected number of backlogged packets of the connection is $w_i - r_i d_i$, where w_i is connection i's window size and r_i is the flow rate. Since TCP Vegas attempts to keep between α and β packets in the switch buffers, if the propagation delay increases, then it should increase the window size to keep the same number of packets in the buffer. Because TCP Vegas relies upon delay estimation, this could impact the performance substantially.

Since the switches in the network do not notify the connections of change in routes, this requires that the source be able to detect any such change in the route. La et el. [12] have suggested the following modification. For the first K packets, it works the same way as TCP Vegas, where K is a prefixed parameter. After the ACK for Kth packet arrives, the source keeps track of the minimum round trip delay of the N consecutive packets. If the minimum round trip time of the last $L \cdot N$ packets is much larger than the current *baseRTT*, the source updates *baseRTT* to the minimum round trip time of the last N packets and resets the congestion window size based on this new *baseRTT*.

The basic idea behind this mechanism is as follows. If the minimum round trip time computed for N packets is consistently much higher than *baseRTT*, then it is likely that the actual propagation delay is larger than the measured *baseRTT*, and it makes sense to increase *baseRTT*. However, it is possible that the increase is due to a persistent congestion in the network. This problem is briefly discussed in section 3.2. Since the increase in delay forces the source to decrease its window size, the round trip delay comes mostly from the propagation delay of the new route. Thus, the minimum round trip delay of the previous N packets is a good estimate of the new propagation delay, as is *baseRTT* for the previous route. The detailed description of the mechanism is given in [12] with simulation results.

3.2 Persistent Congestion

Since TCP Vegas uses *baseRTT* as an estimate of the propagation delay of route, its performance is sensitive to the accuracy of *baseRTT*. Therefore, if the connections overestimate the propagation delay due to incorrect *baseRTT*, it can have a substantial impact on the performance of TCP Vegas. We first consider a scenario where the connections overestimate the propagation delays and possibly drive the system to a persistently congested state.

Suppose that a connection starts when there are many other existing connections, the network

RED HAT 1012
RED HAT V. CAS

is congested and the queues are backed up. Then, due to the queuing delay from other backlogged packets, the packets from the new connection may experience round trip delays that are considerably larger than the actual propagation delay of the path. Hence, the new connection will set the window size to a value such that it believes that its expected number of backlogged packets lies between α and β, when in fact it has many more backlogged packets due to the inaccurate estimation of the propagation delay of the path. This scenario will repeat for each new connection, and it is possible that this causes the system to remain in a persistent congestion. This is exactly the opposite of a desirable scenario. When the network is congested, we do not want the new connections to make the congestion level worse. This same situation may arise with TCP Reno or TCP Tahoe. However, it is more likely to happen with TCP Vegas due to its fine-tuned congestion avoidance mechanism.

Once RED gateways are widely deployed, this problem can be handled by the same mechanism suggested in section 3.1. The intuition behind this solution is as follows. When the network stays in a persistently congested state, the connections interpret the lasting increase in the round trip delay as an increase in the propagation delay and updates their $baseRTT$. This creates a temporary increase in congestion level in the network, and most connections, if not all, back off as they detect the congestion. As connections reduce their window sizes, the congestion level will come down, which allows the connections to estimate the correct $baseRTT$. Once most connections have an accurate measurement of the propagation delay, the congestion level will remain low. Since RED gateways drop packets independently, the synchronization effect will not be a big problem. For more details and simulation results, see [12].

4 Fairness of TCP Vegas

As we have mentioned before, TCP Reno is biased against the connections with long delays [5, 7, 13]. In this section we show that TCP Vegas does not suffer from this delay bias as TCP Reno does by analyzing a simple closed fluid model, and confirm the resulting analysis by simulation in section 6. We assume that the packets are infinitely divisible, i.e., we use a fluid model, and the propagation delay is deterministic.

Consider the simple network in Figure 2. There are two users with possibly different round trip delays, which share a single bottleneck link. User i has round trip delay d_i and exercises window-based flow control. Without loss of generality we assume that the capacity of the bottleneck link is normalized to one.

RED HAT 1012
RED HAT V. CAS

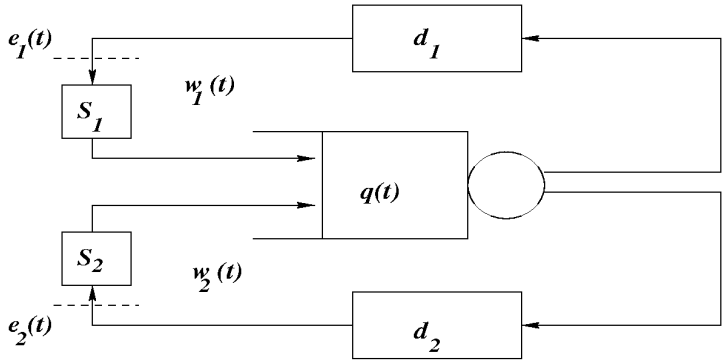


Figure 2: Network with two users

Let $q_i(t)$ denote the user i's queue size at time t, $q(t) = (q_1(t), q_2(t))$, and $z(t) = \sum_{i=1}^{2} q_i(t)$, which is the total backlog at time t. Define $u_i(t) = z(t - u_i(t)) + d_i$, which denotes the round trip delay experienced by user i's packet whose ACK arrives at the source at time t. We assume that the switch adopts the $FIFO$ service discipline and has infinite buffer size. Suppose $e_i(t)$ denotes the rate at which connection i receives ACK as shown in Figure 2 and the capacity is fully utilized, i.e., $e_1(t - d_1) + e_2(t - d_2) = 1$. Users update their window sizes once during each round trip time.

Since the window size is equal to the sum of the amount of fluid in queue and in transit, at time t we have

$$w_i(t) = q_i(t) + \int_t^{t+d_i} e_i(s)ds. \tag{2}$$

Note that the second term on the right hand side is the amount of fluid in transit at time t. Each source computes the expected backlog using the product of the propagation delay and the difference between the expected flow rate and the actual flow rate. Thus, the estimated amount of backlogged fluid computed by connection i is

$$Diff_i = w_i(t) - \frac{d_i}{u_i(t)} \int_{t-u_i(t)}^{t} e_i(s)ds. \tag{3}$$

A heuristic argument for the convergence of window sizes after some time t_0 to a point such that

$$\alpha \leq w_i(t) - \frac{d_i}{u_i(t)} \int_{t-u_i(t)}^{t} e_i(s)ds \leq \beta \tag{4}$$

is given in Appendix. Now suppose that window sizes satisfy equation (4). If we substitute (2) in (4), we have

$$\alpha \leq q_i(t) + \int_t^{t+d_i} e_i(s)ds - \frac{d_i}{u_i(t)} \int_{t-u_i(t)}^{t} e_i(s)ds \leq \beta. \tag{5}$$

RED HAT 1012
551
RED HAT V. CAS

Once the window sizes satisfy (4), the sources do not change the window size. Suppose that the system is quasi-static and $e_i(t)$ is relatively constant. Then, the second and third terms in (5) cancel each other out, and it yields

$$\alpha \leq q_i(t) \leq \beta. \tag{6}$$

Note that under the quasi-static assumption the queue size and the estimated queue size in (3) of connection i do not depend on d_i. Since the server exercises the FIFO service discipline, the flow rate of each connection is proportional to its queue size. Therefore, the throughput of a connection does not depend on its propagation delay, and there is no bias in favor of connections with short delays.

If a connection with a larger delay wants to keep the same amount of fluid in the queue and similar throughput as a connection with a much shorter delay, it needs to have a bigger window size than the connection with a shorter delay because it needs more fluid in transit. This can be easily seen from (2). If both connections have same throughput, then the second term in (2) is directly proportional to the delay of the connection. Therefore, when all connections are receiving similar throughput, the window sizes of the connections should be roughly proportional to their propagation delays.

If we assume that the throughput and the queue size of each connection is relatively constant, then from (2) we get

$$w_i(t) = q_i(t) + e_i(t) \cdot d_i \tag{7}$$

and

$$e_i(t) = \frac{window\ size}{round\ trip\ delay} = \frac{w_i(t)}{z(t) + d_i}. \tag{8}$$

Since we have assumed that the capacity is fully utilized, from (8) we have

$$\frac{w_1(t)}{z(t) + d_1} + \frac{w_2(t)}{z(t) + d_2} = 1. \tag{9}$$

Hence, given the window sizes $w_i(t)$, we can compute $z(t)$ by solving (9). By substituting the solution of (9) in (8) we can compute the throughput of each connection. Plugging in the solution of (8) in (7) yields the queue size of the connection. Thus, under the quasi-static assumption, we can compute the throughput and queue size of the connections, given the window sizes.

The simulation results show that the window sizes converge to a point such that the queue sizes computed from (7) - (9) are between α and β for both connections. Figure 4 shows simulation results and Figure 3 describes how to read the plots in Figure 4. In Figure 3, α_i and β_i lines for

RED HAT 1012
552
RED HAT V. CAS

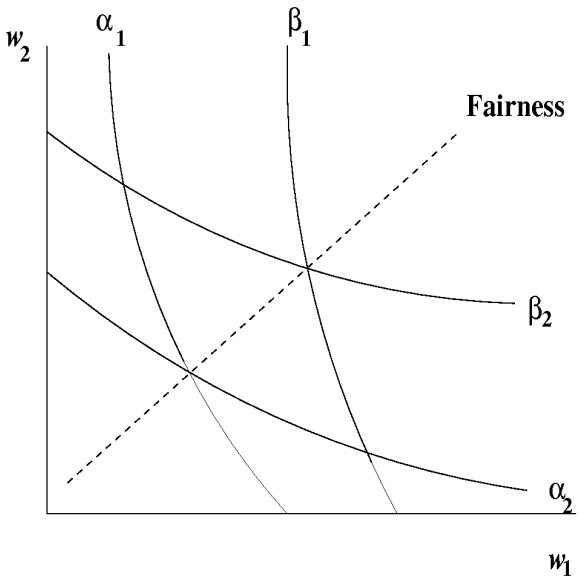


Figure 3: Convergence region

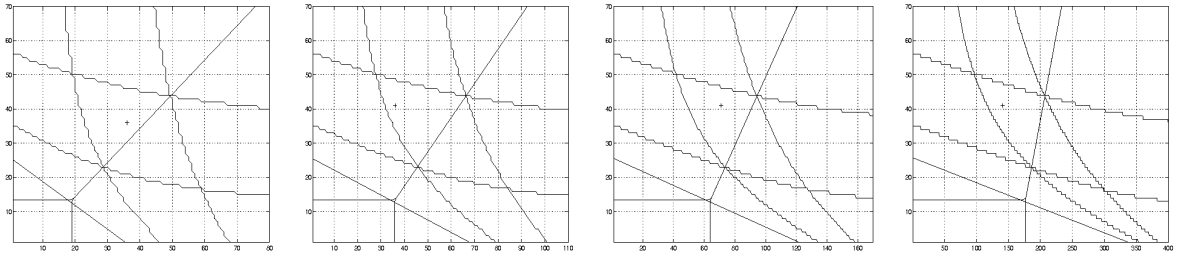


Figure 4: Converging point of TCP-Vegas with different delays.

user i denote the sets of window size pairs $\{(w_1, w_2)|w_i - e_i d_i = \alpha\}$ and $\{(w_1, w_2)|w_i - e_i d_i = \beta\}$, respectively, where e_i is connection i's throughput computed from (7) - (9). The fairness line is the set of window size pairs such that the throughputs of the users are the same, i.e.,$\{(w_1, w_2)|e_1 = e_2\}$. In each plot of Figure 4 '+' denotes the converging point of TCP Vegas. As we can see, in each case it converges to a point (w_1, w_2) that satisfies

$$\alpha \leq q_i(w_1, w_2) \leq \beta, \ \ i = 1, 2, \tag{10}$$

where $q_i(w_1, w_2)$ is the queue size of connection i computed using (7) - (9). The numbers on the X-axis and Y-axis are the window sizes $\times$ 10. The delay of connection 1 (Y-axis) is 12.9 ms in all plots and that of connection 2 (X-axis) is 18.9 ms, 36.9 ms, 66.9 ms, and 186.9 ms, respectively, from left to right.

RED HAT 1012
553
RED HAT V. CAS

5 Incompatibility of TCP Vegas with TCP Reno

There has been some previous work that demonstrates that in many cases TCP Vegas performs better than other implementations of TCP [2, 1]. However, Ahn et al. have observed that when a TCP Vegas user competes with other users that use TCP Reno, it does not receive a fair share of bandwidth due to the conservative congestion avoidance mechanism used by TCP Vegas.

TCP Reno continues to increase the window size until a packet is lost. Packet losses occur mainly due to buffer overflows. This bandwidth estimation mechanism results in a periodic oscillation of window size and buffer-filling behavior of TCP Reno. Thus, while TCP Vegas tries to maintain a smaller queue size, TCP Reno keeps many more packets in the buffer on the average, stealing higher bandwidth.

Let $q_v(t)$ and $q_r(t)$ denote the number of Vegas and Reno packets in the buffer at time t, and B be the buffer size. If we assume $q_v(t) \approx k \leq \beta$, $q_r(t)$ ranges from $[0, B-k]$. Assume that $q_r(t)$ is distributed uniformly on the interval $[0, B-k]$. Then the average queue size of the Reno user $\bar{q}_r(t)$ is approximately $\frac{B-k}{2}$. The ratio of TCP Vegas throughput to TCP Reno throughput is roughly $\frac{2k}{B-k}$. When B is relatively large, which is usually the case, then it is obvious that TCP Reno gets much higher bandwidth than TCP Vegas. We will confirm this analysis with simulation results in the next section.

TCP Reno congestion avoidance scheme is aggressive in the sense that it leaves little room in the buffer for other connections, while TCP Vegas is conservative and tries to occupy little buffer space. When a TCP Vegas connection shares a link with a TCP Reno connection, the TCP Reno connection uses most of the buffer space and the TCP Vegas connection backs off, interpreting this as a sign of network congestion. This may be one of reasons why TCP Vegas is not widely deployed despite its many desirable properties.

6 Simulation Results

In order to verify the observations we have made through the analysis of simple networks, we have run simulations, using real network simulator (ns) developed at Lawrence Berkeley Laboratory [4]. We will discuss the simulation results in this section.

6.1 Fairness of TCP Vegas

RED HAT 1012
RED HAT V. CAS

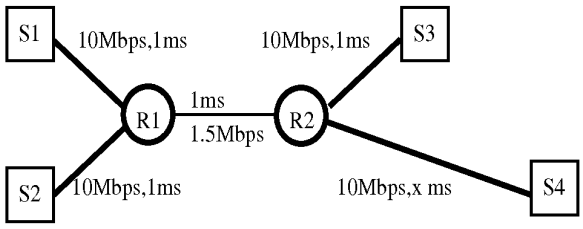


Figure 5: Network Topology

Figure 5 shows the topology of the network that was simulated to demonstrate the fairness of TCP Vegas. In the figure, the circles denote finite-buffer switches and the squares denote the end hosts. Connection 1 transmits packets from S1 to S3, and connection 2 from S2 to S4. The links are labeled with their capacities and propagation delays. We vary the propagation delay of the link that connects R2 and S4, which is denoted by x in Figure 5, in order to see the effect of different delays on the throughputs.

x	w_1	w_2	ACK_1	ACK_2	$\frac{max(ACK_1,ACK_2)}{min(ACK_1,ACK_2)}$
4	3.5	3.5	21,425	16,068	1.33
13	4	7	17,522	19,965	1.14
22	4	7	20,061	17,427	1.15
58	4	13	19,507	17,973	1.09
148	4	30	21,068	16,398	1.29

Table 1: TCP Vegas with varying propagation delays.

x	ACK_1	ACK_2	ACK_1/ACK_2
4	21,100	15,637	1.35
13	25,460	11,785	2.16
22	25,684	11,672	2.20
58	34,429	2,627	13.11
148	35,598	959	37.12

Table 2: TCP Reno with varying propagation delays.

Connections 1 and 2 start at time 0 second and 0.5 seconds, respectively, and the simulation is

RED HAT 1012
RED HAT V. CAS

run for 200 seconds. Tables 1 and 2 show the simulation results. In each table ACK_1 and ACK_2 are the number of packets that have been acknowledged by the corresponding receiver for connections 1 and 2. The last column shows the ratio of $\max(ACK_1, ACK_2)$ to $\min(ACK_1, ACK_2)$.

Tables 1 and 2 clearly show that the delay bias of TCP Reno becomes very noticeable when the delay difference becomes large, while TCP Vegas does not show any such delay bias. The ratio of throughputs under TCP Vegas ranges from 1.09 to 1.33, whereas throughput ratio under TCP Reno increases almost linearly with delay ratio. This is consistent with our analysis given in previous sections and confirms that TCP Vegas does not suffer from the delay bias as TCP Reno does. We can also verify that the expected backlog for TCP Vegas connections at the converging point is between α and β from (7) - (9).

6.2 Incompatibility of Vegas with Reno

In the second part of simulation, we have simulated a network with two connections with same round trip delay of 24 ms, where one is a TCP Reno connection and the other is a TCP Vegas connection, in an attempt to verify the incompatibility of TCP Vegas with TCP Reno. We vary the buffer size at the switches to see how the relative performance of the TCP Vegas connection changes with the buffer size. The simulation results are given in Table 3. The fourth column of the table is the ratio of throughputs, and the last column is the interval computed from $\frac{B-k}{2k}$ as described in section 4, with $k = 1$ and 3, since α and β in TCP Vegas are usually set to 1 and 3, respectively, and k can be any number between α and β.

Buffer Size	ACK of Reno	ACK of Vegas	Reno/Vegas	Interval
4	13,010	24,308	0.535	[0.167, 1.5]
7	16,434	20,903	0.786	[0.667, 3.0]
10	22,091	15,365	1.438	[1.167, 4.5]
15	25,397	12,051	2.107	[2.0, 7.0]
25	30,798	6,621	4.652	[3.667, 12.0]
50	34,443	2,936	11.73	[7.833, 24.5]

Table 3: Throughput of Vegas vs. Reno with drop-tail gateways.

It is clear from Table 3 that TCP Vegas connection does not receive a fair share of bandwidth in the presence of a TCP Reno connection unless the buffer sizes are extremely small. Moreover, the

RED HAT 1012
556
RED HAT V. CAS

ratios of the throughputs fall well within the region computed from the equation $\frac{B-k}{2k}$, confirming the validity of our analysis.

6.3 RED Gateways

The simulation results in the previous subsection suggest a way of making TCP Vegas more competitive in the presence of TCP Reno connections. Note in Table 3 that when the buffer sizes are small, TCP Vegas outperforms TCP Reno. The intuition behind this is as follows. TCP Reno needs some room in the switch buffer for oscillation in order to estimate the available bandwidth. Without the necessary room in the buffer, its performance degrades noticeably. TCP Vegas, on the other hand, quickly adapts to the small buffer size since it requires enough space for only a few packets.

This key observation tells us that Random Early Detection (RED) gateways could be used to make TCP Vegas more compatible with TCP Reno in a competitive environment [7]. RED gateways are characterized by a set of parameters, two of which are *threshold* and *maxthreshold*. RED gateways maintain an estimate of average queue size. When the average queue size exceeds *threshold* but is smaller than *maxthreshold*, they start dropping packets with certain probability that is proportional to the average queue size. If the average queue size exceeds *maxthreshold*, they drop all packets. Hence, if the threshold values are set low enough, they give the connections an impression that the switch buffer size is smaller than it really is. This is discussed in more details in [11].

Threshold	Maxthreshold	ACK of Vegas	ACK of Reno
3	6	61,069	57,774
4	8	54,259	64,305
5	10	50,823	70,020
7	14	44,469	77,202
10	20	34,081	87,379

Table 4: Throughput of Vegas vs. Reno with RED gateways.

The same network used in section 6.2 is simulated again with RED gateways instead of drop-tail gateways. Each simulation is run for 100 seconds, and the buffer size is fixed at 25. The buffer size, however, plays only a minor role if maxthreshold value is smaller than the buffer size. As

RED HAT 1012
557
RED HAT V. CAS

summarized in Table 4, it is easy to see how the threshold values affect the relative performance of TCP Vegas against TCP Reno. As the threshold values increase, TCP Reno throughput increases at the cost of a decrease in TCP Vegas throughput. Also note that the total throughput increases slightly with the threshold values. This is obviously due to the decrease in the number of dropped packets.

7 Conclusions and Further Problems

In this paper we have analyzed the performance of TCP Vegas in comparison with TCP Reno. We have shown that TCP Vegas does lead to a fair allocation of bandwidth and explained some of other characteristics, using the congestion avoidance mechanism adopted by TCP Vegas. We have demonstrated that TCP Vegas does not suffer from the delay bias as TCP Reno does through both analysis and simulations. TCP Vegas achieves better performance than TCP Reno since its bandwidth estimation does not rely on packet losses in order to estimate the available bandwidth in the network. However, when competing with other TCP Reno connections, TCP Vegas gets penalized due to the aggressive nature of TCP Reno.

We are currently investigating the possibility of deploying TCP Vegas, adopting gateway control such as RED gateways. We model the problem both as a single shot game and as an infinitely repeated game. We suspect that we can provide the TCP Reno users with a proper incentive to switch to TCP Vegas by using RED gateways to penalize the users when the average queue sizes fluctuate at the gateways in the presence of TCP Reno users. This problem will be discussed in another paper.

References

[1] J.S. Ahn, P.B. Danzig, Z. Liu, and L. Yan. Evaluation of TCP Vegas: emulation and experiment. *IEEE Transactions on Communications*, 25(4):185–95, Oct 1995.

[2] L.S. Brakmo and L.L. Peterson. Tcp vegas: end to end congestion avoidance on a global internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–80, October 1995.

[3] D. Chiu and R. Jain. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, 17:1–14, 1989.

RED HAT 1012
558
RED HAT V. CAS

[4] K. Fall and S. Floyd. Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. *Computer Communication Review*, 26(3):5–21, July 1996.

[5] S. Floyd and V. Jacobson. Connection with multiple Congested Gateways in packet-Switched Networks,Part1: One-way Traffic". *ACM Computer Communication Review*, 21(5):30–47, August 1991.

[6] S. Floyd and V. Jacobson. On Traffic Phase Effects in packet Switched Gateways. *Internetworking:Research and Experience*, 3(3):115–156, September 1993.

[7] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[8] T. Henderson, E. Sahouria, S McCanne, and R. Katz. Improving Fairness of TCP Congestion Avoidance . In *EECS, UC Berkeley, May '98 (submitted to GLOBECOM '98)*, August 1998.

[9] V. Jacobson. Congestion avoidance and control . *Computer Communication Review*, 18(4):314–29, August 1988.

[10] V. Jacobson. Modified TCP Congestion Avoidance Algorithm. Technical report, April 1990.

[11] Richard J. La, Jeonghoon Mo, Jean Walrand, and Venkat Anantharam. A Case for TCP Vegas and Gateways using Game Theoretic Approach. Available at http://www.path.berkeley.edu/~ hyongla (submitted to INFOCOM '99), July 1998.

[12] Richard J. La, Jean Walrand, and Venkat Anantharam. Issues in TCP Vegas. Available at http://www.path.berkeley.edu/~ hyongla, June 1998.

[13] A. Mankin. Random Drop Congestion Control. In *Proc. SIGCOMM'90*, pages 1–7, Philadelphia, PA, September 1990.

A Convergence of Window Sizes

In this appendix, we give a heuristic argument for the convergence of window sizes to a point (w_1, w_2) such that

$$\alpha \leq Diff_i = w_i(t) - \frac{d_i}{u_i(t)} \int_{t-u_i(t)}^{t} e_i(s)ds \leq \beta \tag{11}$$

559

RED HAT 1012
RED HAT V. CAS

During each round trip time source i computes the estimated backlog

$$Diff_i = w_i(t) - \frac{d_i}{u_i(t)} \int_{t-u_i(t)}^{t} e_i(s)ds, \tag{12}$$

and uses this estimation of the queue size to adjust its window size. We assume that $e_i(t)$ and $q_i(t)$ change slowly and remain steady, and the average converges. This is not an unreasonable assumption because TCP Vegas updates its window size at most by one during one round trip time. Therefore, $e_i(t)$ and $q_i(t)$ do not change too fast. Now suppose that the sources use the average throughput to adjust their window sizes. For the simplicity of notation, we omit the dependency on time t. We have the following equations from (12) and the assumption.

$$Diff_i = w_{i,avg} - e_{i,avg} \cdot d_i \tag{13}$$

$$e_{i,avg} \approx \frac{w_{i,avg}}{z_{avg} + d_i} \tag{14}$$

If we assume that the capacity is fully utilized, we get

$$1 = e_{1,avg} + e_{2,avg} \approx \frac{w_{1,avg}}{z_{avg} + d_1} + \frac{w_{2,avg}}{z_{avg} + d_2}. \tag{15}$$

Hence, given the window sizes $w_{i,avg}$ we can compute an approximate value for z_{avg} from (15). The solution of (15) can be used to compute $e_{i,avg}$ and $Diff_i$ from (13) and (14). The sources update their window size based on $Diff_i$.

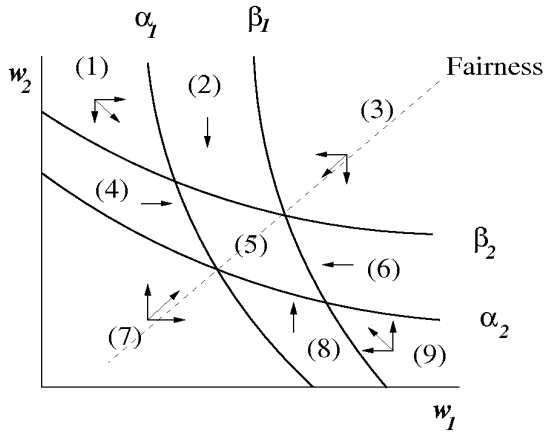


Figure 6: Convergence region of TCP Vegas

Figure 6 illustrates the convergence region of TCP Vegas. In the figure, α_i and β_i lines for user i denote the sets of window size pairs $\{(w_1, w_2)|Diff_i = \alpha\}$ and $\{(w_1, w_2)|Diff_i = \beta\}$, respectively,

RED HAT 1012
560
RED HAT V. CAS

from equations (13) - (15). The fairness line is the set of window size pairs such that the throughputs of the users are the same, i.e.,$\{(w_1, w_2)|e_1 = e_2\}$. User 1 increases its window size in regions (1), (4), and (7), and decreases its window size in regions (3), (6), and (9). Similarly, user 2 increases its window size in regions (7), (8), and (9), and decreases its window size in regions (1), (2), and (3).

Figure 6 is a typical plot of convergence region for two users, and the same geometric picture can be easily extended to a case with more than two users. The only region in the figure where neither user updates its window size is region (5). The arrows in other regions indicate the directions in which the window sizes may get updated.

We will show in Theorem 1 that the window sizes converges to a point in region (5) within a finite amount of time, assuming that the distance between α_i and β_i lines is sufficiently large compared to $\delta > 0$, that is the amount by which users update their window sizes. Suppose $(w_1, w_2) = (w_1(0), w_2(0))$. Let

$$A = \{(w_1 + \delta n_1, w_2 + \delta n_2) \mid 0 \leq w_1 + \delta n_1 \leq w_{1,max}, 0 \leq w_2 + \delta n_2 \leq w_{2,max}, n_1, n_2 \in Z\},$$

where $w_{1,max}$ and $w_{2,max}$ are the maximum congestion window sizes declared by the receivers. Then, A is the set of all possible window size pairs. If $w_{i,max}$ and $w_{2,max}$ are finite, then $N = \mid A \mid$ is finite. We will first state a lemma that will be used in the proof of Theorem 1. Suppose that the distance between α_i and β_i lines is at least $\sqrt{2}\delta$.

Lemma 1 *No window size pair outside region (5) can be visited twice.*

Proof: Suppose that there exists a window size pair that could be visited twice outside region (5). Then, this implies that there exists a loop generated by window size updates as shown in Figure 7. We will now show that no such loop can exist.

In regions (1) through (6), user 2 never increases its window size. Hence, if user 2 updates in regions (1) through (3), then it is not possible to return to the same windows size pair since we have assumed that the distance between α_2 line and β_2 line is larger than δ. Moreover, if user 1 updates its window size in either region (1) or (3), then the previous window size pair cannot be revisited without reaching either region (2) or (5) because user 1 does not update its window size in opposite direction in the same region. However, once the window sizes reach a point in region (2) or (5), then for the same reason given above, it is not possible to return to the previous window size pair because at most only user 2 will update its window size and decrease it. Therefore, it

RED HAT 1012
561
RED HAT V. CAS

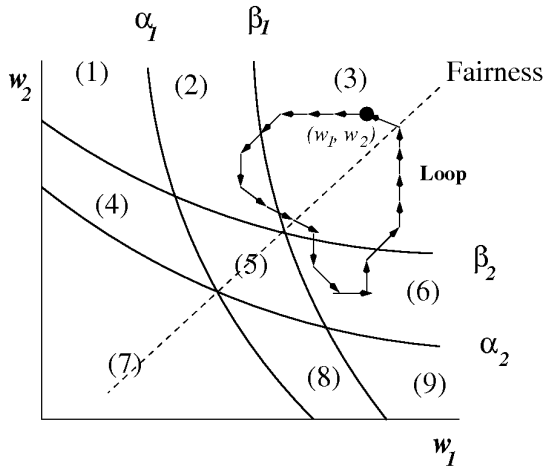


Figure 7: An example of a loop

is not possible for a window size pair in region (1) through (3) to be visited more than once. By symmetry between users 1 and 2, it is not possible to have such a loop for a window size pair in region (3), (6), and (9).

Similarly, since user 2 does not decrease its window size in region (4) through (9), none of window size pairs in regions (7) through (9) can be visited twice. Again by symmetry, the same is true for regions (1), (4), and (7). Therefore, none of the window size pairs outside region (5) can be visited more than once. ■

Theorem 1 *The window sizes converge to a point in region (5) within a finite amount of time, starting from a point in the region bounded by* $w_{1,max}$ *and* $w_{2,max}$.

Proof: From Lemma 1 we know that no window size pair outside region (5) can be visited more than once. Therefore, if $(w_1, w_2) = (w_1(0), w_2(0))$ and $N = \mid A \mid$, then assuming that time between window size updates of each connection is finite, the window sizes converge to a point in (5) because there are at most N-1 other pairs the window size pair could take on before it reaches a point in region (5), and once the window sizes are in region (5), they do not change. ■

RED HAT 1012
562
RED HAT V. CAS

Analysis and Comparison of TCP Reno and Vegas

Jeonghoon Mo, Richard J. La, Venkat Anantharam, and Jean Walrand
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
{jhmo, hyongla, ananth, wlr}@eecs.berkeley.edu

Abstract— **We propose some improvements of TCP Vegas and compare its performance characteristics with TCP Reno. We argue through analysis that TCP Vegas, with its better bandwidth estimation scheme, uses the network resources more efficiently and fairly than TCP Reno. Simulation results are given that support the results of the analysis.**

Keywords—**TCP Reno, TCP Vegas, Congestion Control, Fairness**

I. Introduction

Following a series of congestion collapses starting in October of '86, Jacobson and Karels developed a congestion control mechanism, which was later named TCP Tahoe [9]. Since then, many modifications have been made to the transmission control protocol (TCP) and several different versions of TCP have been implemented [9], [10].

Recently Brakmo *et al.*[2] have proposed a new version of TCP, which is named TCP Vegas, with a fundamentally different congestion avoidance scheme from that of TCP Reno (4.3BSD) and claimed that TCP Vegas achieves 37 to 71 percent higher throughput than TCP Reno. Ahn *et al.*[1] have evaluated the performance of TCP Vegas on their wide area emulator and shown that TCP Vegas does achieve higher efficiency than TCP Reno and causes much fewer packet retransmissions. However, they have observed that TCP Vegas, when competing with other TCP Reno connections, does not receive a fair share of bandwidth.

Floyd has observed that TCP Reno is biased against the connections with longer delays [5], [7] . The intuition behind this behavior is as follows. While a source does not detect any congestion, it continues to increase its window size by one during one round trip time. Obviously connections with a shorter delay can update their window sizes faster than those with longer delays, and thus capture higher bandwidths. Based on this observation, Floyd and Jacobson [6] have proposed a *constant rate adjustment* algorithm. Henderson *et al.*[8] have simulated a variation of this scheme and reported that if the rate of increase of the window size is not excessive, this scheme is not harmful to the other connections that use a different version of TCP. Moreover, as expected, this scheme results in better performance for the connections with longer delays. However, choosing the parameters for such algorithms is still an open problem. Mo and Walrand [15] have demonstrated the existence of fair end-to-end window-based congestion control protocols and proposed an algorithm similar to TCP Vegas, for which convergence results are proved.

In this paper we explain the performance characteristics of TCP Vegas in relation to its queue-based control mechanisms; we demonstrate through both analysis and simulations that TCP Vegas does not suffer from delay bias as TCP Reno does. We show the incompatibility of TCP Reno and TCP Vegas. The rest of the paper is organized as follows. In section 2 we describe the congestion detection and avoidance schemes of TCP Reno and TCP Vegas. Section 3 presents a few problems intrinsic in the original TCP Vegas and explains a possible solution to them. In section 4 we discuss the fairness of TCP Vegas using a simple closed fluid model. Section 5 compares TCP Vegas with TCP Reno and explains the incompatibility between them. We confirm our analysis with simulation results in section 6, and finish with some conclusions and future problems.

II. Congestion Avoidance Mechanism

In order to make efficient use of network bandwidth, TCP controls its flow rate using the feedback from the network. In order to control its flow rate, TCP needs to estimate the available bandwidth in the network using a *bandwidth estimation scheme.* Among many new features implemented in TCP Vegas, the most important difference between TCP Vegas and TCP Reno lies in this bandwidth estimation scheme used to estimate the available bandwidth. As will be shown shortly, TCP Reno views packet losses as a signal of network congestion, while TCP Vegas uses the difference in the expected and actual rates to adjust its window size. This fundamental difference in the bandwidth estimation schemes enables TCP Vegas to utilize the available bandwidth more efficiently.

A. TCP Reno

TCP Reno induces packet losses to estimate the available bandwidth in the network. While there are no packet losses, TCP Reno continues to increase its window size by one during each round trip time. When it experiences a packet loss, it reduces its window size to one half of the current window size. This is called *additive increase and multiplicative decrease.* The additive increase and multiplicative decrease algorithm is based on the results given in [3]. There it is shown that such an algorithm leads to a fair allocation of bandwidth. TCP Reno, however, fails to achieve such fairness because TCP is not a synchronized rate based control scheme, which is necessary for the convergence results of [3].

The congestion avoidance mechanism adopted by TCP Reno causes a periodic oscillation in the window size due to the constant update of the window size. This oscillation in the window size leads to an oscillation in the round trip delay of the packets. This oscillation results in larger delay jitter and an inefficient use of the available bandwidth due to many retransmissions of the same packets after packet drops occur.

0-7803-5417-6/99/$10.00 ©1999 IEEE.

RED HAT 1012
RED HAT V. CAS

B. TCP Vegas

TCP Vegas adopts a more sophisticated bandwidth estimation scheme. It uses the difference between expected and actual flows rates to estimate the available bandwidth in the network. The idea is that when the network is not congested, the actual flow rate will be close to the expected flow rate. Otherwise, the actual flow rate will be smaller than the expected flow rate. TCP Vegas, using this difference in flow rates, estimates the congestion level in the network and updates the window size accordingly. Note that this difference in the flow rates can be easily translated into the difference between the window size and the number of acknowledged packets during the round trip time, using the equation

$$Diff = (Expected - Actual)\ BaseRTT, \qquad (1)$$

where *Expected* is the expected rate, *Actual* is the actual rate, and *BaseRTT* is the minimum round trip time. The details of the algorithm are as follows:

1. First, the source computes the expected flow rate *Expected* = $\frac{CWND}{BaseRTT}$, where *CWND* is the current window size and *BaseRTT* is the minimum round trip time.
2. Second, the source estimates the current flow rate by using the actual round trip time according to *Actual* = $\frac{CWND}{RTT}$, where RTT is the actual round trip time of a packet.
3. The source, using the expected and actual flow rates, computes the estimated backlog in the queue from *Diff* = *(Expected-Actual) BaseRTT*.
4. Based on *Diff*, the source updates its window size as follows.

$$CWND = \begin{cases} CWND + 1 & if\ Diff < \alpha \\ CWND - 1 & if\ Diff > \beta \\ CWND & otherwise \end{cases}$$

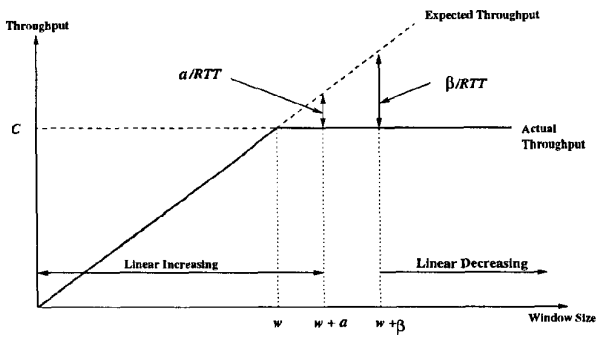


Fig. 1. Window control of TCP Vegas

Figure 1 illustrates the behavior of TCP Vegas. Consider a simple network with one connection and one link with capacity C. Let *BaseRTT* be the minimum round trip delay. The throughput of this connection is $\frac{window}{BaseRTT}$ when $window < C \times$ *BaseRTT*. In the figure, w corresponds to the window size where $window = C \times BaseRTT$. When *window* $> w$, a queue starts to build up and *(Expected - Actual)* > 0. TCP Vegas increases the window size by one during the next round trip time if *window* $< w + \alpha$ and decreases the window size by one if *window* $> w + \beta$. Otherwise, it leaves the window size unchanged. In the figure, *Diff* is user i's estimated backlogged queue size in the link. This will be explained in more details in section 3. TCP Vegas tries to keep at least α packets but no more than β packets in the queues. The reason behind this is that TCP Vegas attempts to detect and utilize the extra bandwidth whenever it becomes available without congesting the network. Thus, when there is only one connection, the window size of TCP Vegas converges to a point that lies between $w + \alpha$ and $w + \beta$. Note that this mechanism is fundamentally different from that used by TCP Reno. TCP Reno always updates its window size to guarantee full utilization of available bandwidth, leading to constant packet losses, whereas TCP Vegas does not cause any oscillation in window size once it converges to an equilibrium point.

III. TCP Vegas

TCP Vegas has a few problems that have not been discussed before, which could have a serious impact on its performance. One of the problems is the issue of rerouting. Since TCP Vegas uses an estimate of the propagation delay, *baseRTT*, to adjust its window size, it is very important for a TCP Vegas connection to have an accurate estimate of this quantity. Rerouting a path may change the propagation delay of the connection, and this could result in a substantial decrease in throughput. Another important issue is the stability of TCP Vegas. Since each TCP Vegas connection attempts to keep a few packets in the network, when the estimation of the propagation delay is off, this could lead the connections to inadvertently keep many more packets in the network, causing a persistent congestion. La *et al.*[13] have investigated these issues. We summarize the results here. For more details, refer to [13].

A. Rerouting

TCP Vegas, as first suggested by Brakmo *et al.*[2], does not have any mechanism that handles the rerouting of connection. In this section, we show that such a lack of mechanisms could lead to strange behavior for TCP Vegas connections. Recall that in TCP Vegas *BaseRTT* is the smallest round trip delay, which is an estimate of the propagation delay of the path .

First, if the route of a connection is changed by a switch, then without an explicit signal from the switch, the end host cannot directly detect it. If the new route has a shorter propagation delay, this does not cause any serious problem for TCP Vegas because most likely some packets will experience shorter round trip delay and *BaseRTT* will be updated. On the other hand, if the new route for the connection has a longer propagation delay, the connection will not be able to tell whether the increase in the round trip delay is due to a congestion in the network or a change in the route. Without this knowledge the end host will interpret the increase in the round trip delay as a sign of congestion in the network and decrease the window size. This is, however, the opposite of what the source should do. When the propagation delay of connection i is d_i, the expected number of backlogged packets of the connection is $w_i - r_i d_i$, where w_i is connection i's window size and r_i is the flow rate. Since TCP Vegas attempts to keep between α and β packets in the switch buffers, if the propagation delay increases, then it should increase the window size to keep the same number of packets in the buffer. Because

RED HAT 1012
RED HAT V. CAS

TCP Vegas relies upon delay estimation, the rerouting could impact the performance substantially.

Since switches in the network do not notify the connections of changes in routes, it is required for sources of the connections to be able to detect such changes. La *et al.*[13] have suggested the following modification. For the first K packets, where K is a prefixed parameter, the modified protocol works the same way as TCP Vegas. After the ACK for the Kth packet arrives, the source keeps track of the minimum round trip delay of N consecutive packets. If the minimum round trip time of the last $L \cdot N$ packets is much larger than the current *baseRTT*, the source updates *baseRTT* to the minimum round trip time of the last N packets and resets the congestion window size based on this new *baseRTT*.

The basic idea behind this mechanism is as follows. If the minimum round trip time computed for N packets is consistently much higher than *baseRTT*, then it is likely that the actual propagation delay is larger than the measured *baseRTT*, and it makes sense to increase *baseRTT*. However, it is possible that the increase is due to a persistent congestion in the network. This problem is briefly discussed in section III-B. Since the increase in delay forces the source to decrease its window size, the round trip delay comes mostly from the propagation delay of the new route. Thus, the minimum round trip delay of the previous N packets is a good estimate of the new propagation delay, as is *baseRTT* for the previous route. The detailed description of the mechanism is given in [13] with simulation results.

B. Persistent Congestion

Since TCP Vegas uses *baseRTT* as an estimate of the propagation delay of route, its performance is sensitive to the accuracy of *baseRTT*. Therefore, if the connections overestimate *baseRTT* due to a persistent congestion, the overestimation can have a substantial impact on the performance of TCP Vegas. We consider a scenario where the connections overestimate the propagation delays and possibly drive the system to a persistently congested state.

Suppose that a connection starts when there are many other existing connections, the network is congested, and the queues are backed up. Then, due to the queuing delay from other backlogged packets, the packets from the new connection may experience round trip delays that are considerably larger than the actual propagation delay of the path. Hence, the new connection will set the window size to a value such that it believes that its expected number of backlogged packets lies between α and β, when in fact it has many more backlogged packets due to the inaccurate estimation of the propagation delay of the path. This scenario will repeat for each new connection, and it is possible that this causes the system to remain in a state of persistent congestion. This is exactly the opposite of what is desirable. When the network is congested, we do not want the new connections to make the congestion level worse. This same situation may arise with TCP Reno or TCP Tahoe. However, it is more likely to happen with TCP Vegas due to its fine-tuned congestion avoidance mechanism.

Once RED gateways are widely deployed, this problem can be handled by the same mechanism suggested in section III.A. The intuition behind this solution is as follows. When the network stays in a persistently congested state, the connections interpret the lasting increase in the round trip delay as an increase in the propagation delay and update their *baseRTT*. This creates a temporary increase in congestion level in the network, and most connections, if not all, back off as they detect the congestion. As connections reduce their window sizes, the congestion level will come down, which allows the connections to estimate the correct *baseRTT*. Once most connections have an accurate measurement of the propagation delay, the congestion level will remain low. Since RED gateways drop packets independently, the synchronization effect will not be a big problem. For more details and simulation results, see [13].

IV. Fairness of TCP Vegas

As we have mentioned before, TCP Reno is biased against the connections with long delays [5], [7], [14]. In this section we show that TCP Vegas does not suffer from this delay bias as TCP Reno does by analyzing a simple closed fluid model, and confirm the resulting analysis by simulation in section 6. We assume that the packets are infinitely divisible, i.e., we use a fluid model, and the propagation delay is deterministic.

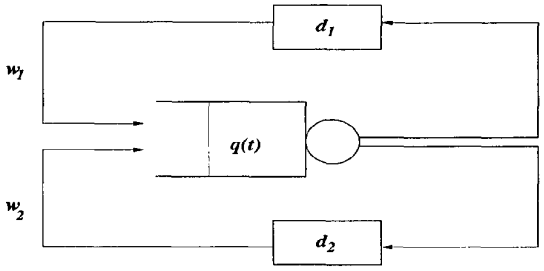


Fig. 2. Network with two users

Consider the simple network in Figure 2. There are two users with possibly different round trip delays, which share a single bottleneck link. User i has round trip delay d_i and exercises window-based flow control. Without loss of generality we assume that the capacity of the bottleneck link is normalized to one.

Let $q_i(t)$ denote the user i's queue size at time t, $q(t) = (q_1(t), q_2(t))$, and $z(t) = \sum_{i=1}^{2} q_i(t)$, which is the total backlog at time t. Define $u_i(t) = z(t - u_i(t)) + d_i$, which denotes the round trip delay experienced by user i's packet whose ACK arrives at the source at time t. We assume that the switch adopts the $FIFO$ service discipline and has infinite buffer size. Suppose $e_i(t)$ denotes the rate at which connection i receives ACK, as shown in Figure 2, and the capacity is fully utilized, i.e., $e_1(t - d_1) + e_2(t - d_2) = 1$. Users update their window sizes once during each round trip time.

Since the window size is equal to the sum of the amount of fluid in queue and in transit, at time t we have

$$w_i(t) = q_i(t) + \int_t^{t+d_i} e_i(s)ds. \qquad (2)$$

Note that the second term on the right hand side is the amount of fluid in transit at time t. Each source computes the expected

RED HAT 1012
RED HAT V. CAS

backlog using the product of the propagation delay and the difference between the expected flow rate and the actual flow rate. Thus, the estimated amount of backlogged fluid computed by connection i is

$$Diff_i = w_i(t) - \frac{d_i}{u_i(t)} \int_{t-u_i(t)}^{t} e_i(s)ds. \quad (3)$$

A heuristic argument for the convergence of window sizes after some time t_0 to a point such that

$$\alpha \leq w_i(t) - \frac{d_i}{u_i(t)} \int_{t-u_i(t)}^{t} e_i(s)ds \leq \beta \quad (4)$$

is given in the Appendix. Now suppose that window sizes satisfy equation (4). If we substitute (2) in (4), we have

$$\alpha \leq q_i(t) + \int_{t}^{t+d_i} e_i(s)ds - \frac{d_i}{u_i(t)} \int_{t-u_i(t)}^{t} e_i(s)ds \leq \beta. \quad (5)$$

Once the window sizes satisfy (4), the sources do not change the window size. Suppose that the system is quasi-static and $e_i(t)$ is relatively constant. Then, the second and third terms in (5) cancel each other out, and it yields

$$\alpha \leq q_i(t) \leq \beta. \quad (6)$$

Note that under the quasi-static assumption the queue size and the estimated queue size in (3) of connection i do not depend on d_i. Since the server exercises the FIFO service discipline, the flow rate of each connection is proportional to its queue size. Therefore, the throughput of a connection does not depend on its propagation delay, and there is no bias in favor of connections with short delays. Furthermore, if α and β are set to 1 and 3, respectively, the worst case ratio of throughput among the connections is at most 3.

If a connection with a larger delay wants to keep the same amount of fluid in the queue and similar throughput as a connection with a much shorter delay, it needs to have a bigger window size than the connection with a shorter delay because it needs more fluid in transit. This can be easily seen from (2). If both connections have same throughput, then the second term in (2) is directly proportional to the delay of the connection. Therefore, when all connections are receiving similar throughput, the window sizes of the connections should be roughly proportional to their propagation delays.

If we assume that the throughput and the queue size of each connection is relatively constant, then from (2) we get

$$w_i(t) = q_i(t) + e_i(t) \cdot d_i \quad (7)$$

and

$$e_i(t) = \frac{window\ size}{round\ trip\ delay} = \frac{w_i(t)}{z(t) + d_i}. \quad (8)$$

Since we have assumed that the capacity is fully utilized, from (8) we have

$$\frac{w_1(t)}{z(t) + d_1} + \frac{w_2(t)}{z(t) + d_2} = 1. \quad (9)$$

Hence, given the window sizes $w_i(t)$, we can compute $z(t)$ by solving (9). By substituting the solution of (9) in (8) we can compute the throughput of each connection. Plugging the solution of (8) in (7) yields the queue size of the connection. Thus, under the quasi-static assumption, we can compute the throughput and queue size of the connections, given the window sizes.

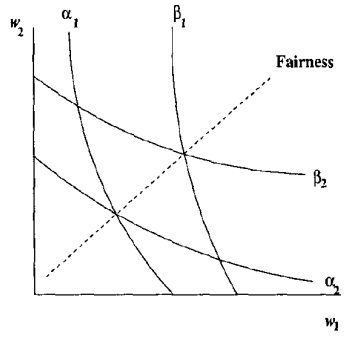


Fig. 3. Convergence Region

The simulation results show that the window sizes converge to a point such that the queue sizes computed from (7) - (9) are between α and β for both connections. Figure 4 shows simulation results and Figure 3 describes how to read the plots in Figure 4. In Figure 3, α_i and β_i lines for user i denote the sets of window size pairs $\{(w_1, w_2)|w_i - e_i d_i = \alpha\}$ and $\{(w_1, w_2)|w_i - e_i d_i = \beta\}$, respectively, where e_i is connection i's throughput computed from (7) - (9). The fairness line is the set of window size pairs such that the throughputs of the users are the same, i.e.,$\{(w_1, w_2)|e_1 = e_2\}$. In each plot of Figure 4 '+' denotes the converging point of TCP Vegas. As we can see, in each case it converges to a point (w_1, w_2) that satisfies

$$\alpha \leq q_i(w_1, w_2) \leq \beta,\ i = 1, 2, \quad (10)$$

where $q_i(w_1, w_2)$ is the queue size of connection i computed using (7) - (9). The numbers on the X-axis and Y-axis are the window sizes $\times$ 10. The delay of connection 1 (Y-axis) is 12.9 ms in all plots and that of connection 2 (X-axis) is 18.9 ms, 36.9 ms, 66.9 ms, and 186.9 ms, respectively, from left to right and top to bottom.

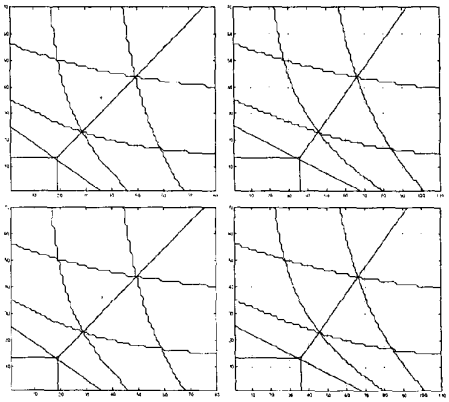


Fig. 4. Converging point of TCP-Vegas with different delays.

RED HAT 1012
RED HAT V. CAS

V. Incompatibility of TCP Vegas with TCP Reno

There has been some previous work that demonstrates that in many cases TCP Vegas performs better than other implementations of TCP [2], [1]. However, Ahn *et al.*have observed that when a TCP Vegas user competes with other users that use TCP Reno, it does not receive a fair share of bandwidth due to the conservative congestion avoidance mechanism used by TCP Vegas.

TCP Reno continues to increase the window size until a packet is lost. Packet losses occur mainly due to buffer overflows. This bandwidth estimation mechanism results in a periodic oscillation of window size and buffer-filling behavior of TCP Reno. Thus, while TCP Vegas tries to maintain a smaller queue size, TCP Reno keeps many more packets in the buffer on the average, stealing higher bandwidth.

Let $q_v(t)$ and $q_r(t)$ denote the number of Vegas and Reno packets in the buffer at time t, and B be the buffer size. If we assume $q_v(t) \approx k \leq \beta$, $q_r(t)$ ranges from $[0, B-k]$. Assume that $q_r(t)$ is distributed uniformly on the interval $[0, B-k]$. Then the average queue size of the Reno user $\bar{q}_r(t)$ is approximately $\frac{B-k}{2}$. The ratio of TCP Vegas throughput to TCP Reno throughput is roughly $\frac{2k}{B-k}$. When B is relatively large, which is usually the case, then it is obvious that TCP Reno gets much higher bandwidth than TCP Vegas. We will confirm this analysis with simulation results in the next section.

The TCP Reno congestion avoidance scheme is aggressive in the sense that it leaves little room in the buffer for other connections, while TCP Vegas is conservative and tries to occupy little buffer space. When a TCP Vegas connection shares a link with a TCP Reno connection, the TCP Reno connection uses most of the buffer space and the TCP Vegas connection backs off, interpreting this as a sign of network congestion. This may be one of reasons why TCP Vegas is not widely deployed despite its many desirable properties. This same problem exists with the algorithm proposed by Mo and Walrand [15].

Although TCP Reno connections receive higher throughput than TCP Vegas connections when the buffer sizes are large, their packets experience unnecessarily long delays and jitter due to the large average queue sizes. TCP Vegas, on the other hand, maintains very small backlogs in the network, which results in short delays, and once the system reaches an equilibrium, the delay jitter becomes very small as well.

VI. Simulation Results

In order to verify the observations we have made through the heuristic analysis of simple networks, we have run simulations, using the network simulator (ns) developed at Lawrence Berkeley Laboratory [4]. We will discuss the simulation results in this section.

A. Fairness of TCP Vegas

Figure 5 shows the topology of the network that was simulated to demonstrate the fairness of TCP Vegas. In the figure, the circles denote finite-buffer switches and the squares denote the end hosts. Connection 1 transmits packets from S1 to S3, and connection 2 from S2 to S4. The links are labeled with their capacities and propagation delays. The packet size is fixed at

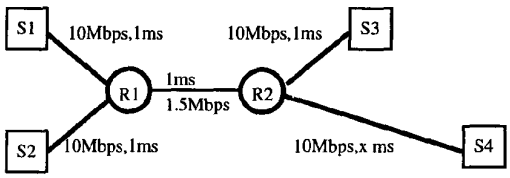


Fig. 5. Network Topology

1,000 bytes. We vary the propagation delay of the link that connects R2 and S4, which is denoted by x in Figure 5, in order to see the effect of different delays on the throughputs.

x	w_1	w_2	ACK_1	ACK_2	$\frac{max(ACK_1,ACK_2)}{min(ACK_1,ACK_2)}$
4	3.5	3.5	21,425	16,068	1.33
13	4	7	17,522	19,965	1.14
22	4	7	20,061	17,427	1.15
58	4	13	19,507	17,973	1.09
148	4	30	21,068	16,398	1.29

TABLE I
TCP Vegas with varying propagation delays.

x	ACK_1	ACK_2	ACK_1/ACK_2
4	21,100	15,637	1.35
13	25,460	11,785	2.16
22	25,684	11,672	2.20
58	34,429	2,627	13.11
148	35,598	959	37.12

TABLE II
TCP Reno with varying propagation delays.

Connections 1 and 2 start at time 0 and 0.5 seconds, respectively, and the simulation is run for 200 seconds. Tables I and II show the simulation results. In each table ACK_1 and ACK_2 are the number of packets that have been acknowledged by the corresponding receiver for connections 1 and 2. The last column shows the ratio of $\max(ACK_1, ACK_2)$ to $\min(ACK_1, ACK_2)$.

Tables I and II clearly show that the delay bias of TCP Reno becomes very noticeable when the delay difference becomes large, while TCP Vegas does not show any such delay bias. The ratio of throughputs under TCP Vegas ranges from 1.09 to 1.33, whereas throughput ratio under TCP Reno increases almost linearly with delay ratio. This is consistent with our analysis given in previous sections and confirms that TCP Vegas does not suffer from the delay bias as TCP Reno does. We can also verify that the expected backlog for TCP Vegas connections at the converging point is between α and β from (7) - (9).

B. Incompatibility of Vegas with Reno

In the second part of simulation, we have simulated a network with two connections with same round trip delay of 24 ms, where one is a TCP Reno connection and the other is a TCP

RED HAT 1012
RED HAT V. CAS

Vegas connection, in an attempt to verify the incompatibility of TCP Vegas with TCP Reno. The packet size is again fixed at 1,000 bytes. We vary the buffer size at the switches to see how the relative performance of the TCP Vegas connection changes with the buffer size. The simulation results are given in Table III. The fourth column of the table is the ratio of throughputs, and the last column is the interval computed from $\frac{B-k}{2k}$ as described in section 4, with $k = 1$ and 3, since α and β in TCP Vegas are usually set to 1 and 3, respectively, and k can be any number between α and β.

Buffer	ACK(R)	ACK(V)	Reno/Vegas	Interval
4	13,010	24,308	0.535	[0.167, 1.5]
7	16,434	20,903	0.786	[0.667, 3.0]
10	22,091	15,365	1.438	[1.167, 4.5]
15	25,397	12,051	2.107	[2.0, 7.0]
25	30,798	6,621	4.652	[3.667, 12.0]
50	34,443	2,936	11.73	[7.833, 24.5]

TABLE III

THROUGHPUT OF VEGAS VS. RENO WITH DROP-TAIL GATEWAYS (R - RENO AND V - VEGAS). BUFFER SIZES ARE GIVEN IN PACKETS.

It is clear from Table III that TCP Vegas connection does not receive a fair share of bandwidth in the presence of a TCP Reno connection unless the buffer sizes are extremely small. Moreover, the ratios of the throughputs fall well within the region computed from the equation $\frac{B-k}{2k}$, suggested by our analysis.

C. RED Gateways

The simulation results in the previous subsection suggest a way of making TCP Vegas more competitive in the presence of TCP Reno connections. Note in Table III that when the buffer sizes are small, TCP Vegas outperforms TCP Reno. The intuition behind this is as follows. TCP Reno needs some room in the switch buffer for oscillation in order to estimate the available bandwidth. Without the necessary room in the buffer, its performance degrades noticeably. TCP Vegas, on the other hand, quickly adapts to the small buffer size since it requires enough space for only a few packets.

This key observation tells us that Random Early Detection (RED) gateways could be used to make TCP Vegas more compatible with TCP Reno in a competitive environment [7]. RED gateways are characterized by a set of parameters, two of which are *threshold* and *maxthreshold*. RED gateways maintain an estimate of average queue size. When the average queue size exceeds *threshold* but is smaller than *maxthreshold*, they start dropping packets with certain probability that is proportional to the average queue size. If the average queue size exceeds *maxthreshold*, they drop all incoming packets. Hence, if the threshold values are set low enough, they give the connections an impression that the switch buffer size is smaller than it really is. This is discussed in more detail in [12].

The same network used in section VI.B is simulated again with RED gateways instead of drop-tail gateways. Each simulation is run for 100 seconds, and the buffer size is fixed at 25 packets. The buffer size, however, plays only a minor role if *maxthreshold* is smaller than the buffer size. As summarized in Table IV, it is easy to see how the threshold values affect the relative performance of TCP Vegas against TCP Reno. As the threshold values increase, TCP Reno throughput increases at the cost of a decrease in TCP Vegas throughput. Also note that the total throughput increases slightly with the threshold values. This is obviously due to the decrease in the number of dropped packets.

Threshold	Maxthreshold	ACK of Vegas	ACK of Reno
3	6	61,069	57,774
4	8	54,259	64,305
5	10	50,823	70,020
7	14	44,469	77,202
10	20	34,081	87,379

TABLE IV

THROUGHPUT OF VEGAS VS. RENO WITH RED GATEWAYS.

VII. CONCLUSIONS AND FURTHER PROBLEMS

In this paper we have analyzed the performance of TCP Vegas in comparison with TCP Reno. We have shown that TCP Vegas does lead to a fair allocation of bandwidth and explained some of its other characteristics. We have demonstrated through both analysis and simulations that TCP Vegas does not suffer from the delay bias as TCP Reno does. TCP Vegas achieves better performance than TCP Reno since its bandwidth estimation does not rely on packet losses in order to estimate the available bandwidth in the network. However, when competing with other TCP Reno connections, TCP Vegas gets penalized due to the aggressive nature of TCP Reno.

We are currently investigating the possibility of deploying TCP Vegas, adopting gateway control such as deficit round-robin (DRR) gateways. We first model the problem as a single shot game and explain the reason that TCP Vegas is difficult to deploy in the current Internet without providing a sufficient incentive. We suspect that we can provide the TCP Reno users with a proper incentive to switch to TCP Vegas by using DRR gateways to provide the necessary isolation among the users and thus inducing the users to cooperate with the network for their own benefit [11].

REFERENCES

[1] J.S. Ahn, P.B. Danzig, Z. Liu, and L. Yan. Evaluation of TCP Vegas: emulation and experiment. *IEEE Transactions on Communications*, 25(4):185–95, Oct 1995.

[2] L.S. Brakmo and L.L. Peterson. Tcp vegas: end to end congestion avoidance on a global internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–80, October 1995.

[3] D. Chiu and R. Jain. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, 17:1–14, 1989.

[4] K. Fall and S. Floyd. Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. *Computer Communication Review*, 26(3):5–21, July 1996.

[5] S. Floyd and V. Jacobson. Connection with multiple Congested Gateways in packet-Switched Networks,Part1: One-way Traffic". *ACM Computer Communication Review*, 21(5):30–47, August 1991.

[6] S. Floyd and V. Jacobson. On Traffic Phase Effects in packet Switched Gateways. *Internetworking:Research and Experience*, 3(3):115–156, September 1993.

RED HAT 1012
RED HAT V. CAS

[7] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
[8] T. Henderson, E. Sahouria, S McCanne, and R. Katz. Improving Fairness of TCP Congestion Avoidance . In *Globecom '98*, 1998.
[9] V. Jacobson. Congestion avoidance and control . *Computer Communication Review*, 18(4):314–29, August 1988.
[10] V. Jacobson. Modified TCP Congestion Avoidance Algorithm. Technical report, April 1990.
[11] Richard J. La and Venkat Anantharam. A Case Study for TCP Vegas and Deficit Round-Robin Gateways. Available at http://www.path.berkeley.edu/~ hyongla, December 1998.
[12] Richard J. La, Jeonghoon Mo, Jean Walrand, and Venkat Anantharam. A Case for TCP Vegas and Gateways using Game Theoretic Approach. Available at http://www.path.berkeley.edu/~ hyongla, July 1998.
[13] Richard J. La, Jean Walrand, and Venkat Anantharam. Issues in TCP Vegas. available at http://www.path.berkeley.edu/~ hyongla, July 1998.
[14] A. Mankin. Random Drop Congestion Control. In *Proc. SIGCOMM'90*, pages 1–7, Philadelphia,PA, September 1990.
[15] Jeonghoon Mo and Jean Walrand. Fair end to end window-based congestion control. In *Proceedings of SPIE '98 Performance and Control of Network Systems II*, pages 55–63, Boston,MA, November 1998.

APPENDIX

I. CONVERGENCE OF WINDOW SIZES

In this appendix, we give a heuristic argument for the convergence of window sizes to a point (w_1, w_2) such that

$$\alpha \leq Diff_i = w_i(t) - \frac{d_i}{u_i(t)} \int_{t-u_i(t)}^{t} e_i(s)ds \leq \beta \qquad (11)$$

During each round trip time source i computes the estimated backlog

$$Diff_i = w_i(t) - \frac{d_i}{u_i(t)} \int_{t-u_i(t)}^{t} e_i(s)ds, \qquad (12)$$

and uses this estimate of the queue size to adjust its window size. We assume that $e_i(t)$ and $q_i(t)$ change slowly and remain steady, and the average converges. This is not an unreasonable assumption because TCP Vegas updates its window size at most by one during one round trip time. Therefore, $e_i(t)$ and $q_i(t)$ do not change too fast. Now suppose that the sources use the average throughput to adjust their window sizes. For the simplicity of notation, we omit the dependency on time t. We have the following equations from (12) and the assumption.

$$Diff_i = w_{i,avg} - e_{i,avg} \cdot d_i \qquad (13)$$

$$e_{i,avg} \approx \frac{w_{i,avg}}{z_{avg} + d_i} \qquad (14)$$

If we assume that the capacity is fully utilized, we get

$$1 = e_{1,avg} + e_{2,avg} \approx \frac{w_{1,avg}}{z_{avg} + d_1} + \frac{w_{2,avg}}{z_{avg} + d_2}. \qquad (15)$$

Hence, given the window sizes $w_{i,avg}$ we can compute an approximate value for z_{avg} from (15). The solution of (15) can be used to compute $e_{i,avg}$ and $Diff_i$ from (13) and (14). The sources update their window size based on $Diff_i$.

Figure 6 illustrates the convergence region of TCP Vegas. In the figure, α_i and β_i lines for user i denote the sets of window size pairs $\{(w_1, w_2)|Diff_i = \alpha\}$ and $\{(w_1, w_2)|Diff_i = \beta\}$, respectively, from equations (13) - (15). The fairness line is the set of window size pairs such that the throughputs of the users are

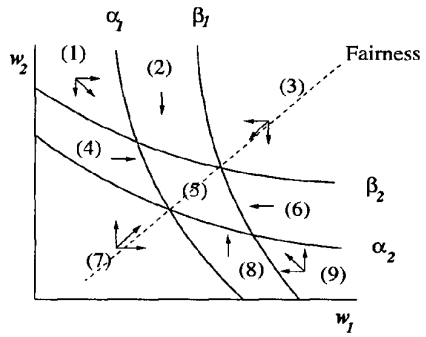


Fig. 6. Convergence region of TCP Vegas

the same, i.e.,$\{(w_1, w_2)|e_1 = e_2\}$. User 1 increases its window size in regions (1), (4), and (7), and decreases its window size in regions (3), (6), and (9). Similarly, user 2 increases its window size in regions (7), (8), and (9), and decreases its window size in regions (1), (2), and (3).

Figure 6 is a typical plot of convergence region for two users, and the same geometric picture can be easily extended to a case with more than two users. The only region in the figure where neither user updates its window size is region (5). The arrows in other regions indicate the directions in which the window sizes may get updated.

We will show in Theorem 1 that the window sizes converges to a point in region (5) within a finite amount of time, assuming that the distance between α_i and β_i lines is sufficiently large compared to $\delta > 0$, the amount by which users update their window sizes. Suppose $(w_1, w_2) = (w_1(0), w_2(0))$. Let

$$A = \{(w_1 + \delta n_1, \quad w_2 + \delta n_2) \mid 0 \leq w_1 + \delta n_1 \leq w_{1,max}, \\ 0 \leq w_2 + \delta n_2 \leq w_{2,max}, n_1, n_2 \in Z\},$$

where $w_{1,max}$ and $w_{2,max}$ are the maximum congestion window sizes declared by the receivers. Then, A is the set of all possible window size pairs. If $w_{i,max}$ and $w_{2,max}$ are finite, then $N = \mid A \mid$ is finite. We will first state a lemma that will be used in the proof of Theorem 1. Suppose that the distance between the α_i and β_i lines is at least $\sqrt{2}\delta$.

Lemma 1: No window size pair outside region (5) can be visited twice.

Proof: Suppose that there exists a window size pair that could be visited twice outside region (5). Then, this implies that there exists a loop generated by window size updates as shown in Figure 7. We will now show that no such loop can exist.

In regions (1) through (6), user 2 never increases its window size. Hence, if user 2 updates in regions (1) through (3), then it is not possible to return to the same windows size pair since we have assumed that the distance between α_2 line and β_2 line is larger than δ. Moreover, if user 1 updates its window size in either region (1) or (3), then the previous window size pair cannot be revisited without reaching either region (2) or (5) because user 1 does not update its window size in the opposite direction in the same region. However, once the window sizes reach a point in region (2) or (5), then for the same reason given above, it is not possible to return to the previous window size pair because

RED HAT 1012
RED HAT V. CAS

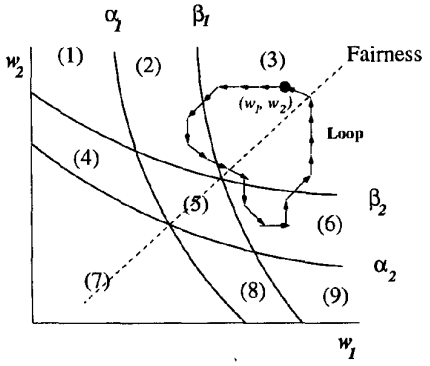


Fig. 7. An example of a loop

at most only user 2 will update its window size and decrease it. Therefore, it is not possible for a window size pair in region (1) through (3) to be visited more than once. By symmetry between users 1 and 2, it is not possible to have such a loop for a window size pair in region (3), (6), and (9).

Similarly, since user 2 does not decrease its window size in region (4) through (9), none of window size pairs in regions (7) through (9) can be visited twice. Again by symmetry, the same is true for regions (1), (4), and (7). Therefore, none of the window size pairs outside region (5) can be visited more than once. ■

Theorem 1: The window sizes converge to a point in region (5) within a finite amount of time, starting from a point in the region bounded by $w_{1,max}$ and $w_{2,max}$.

Proof: From Lemma 1 we know that no window size pair outside region (5) can be visited more than once. Therefore, if $(w_1, w_2) = (w_1(0), w_2(0))$ and $N = \mid A \mid$, then assuming that the time between window size updates of each connection is finite, the window sizes converge to a point in (5) because there are at most N-1 other pairs the window size pair could take on before it reaches a point in region (5), and once the window sizes are in region (5), they do not change. ■

RED HAT 1012
RED HAT V. CAS

Dynamic Parallel Access to Replicated Content in the Internet

Pablo Rodriguez and Ernst W. Biersack, *Member, IEEE*

***Abstract*—Popular content is frequently replicated in multiple servers or caches in the Internet to offload origin servers and improve end-user experience. However, choosing the best server is a nontrivial task and a bad choice may provide poor end user experience. In contrast to retrieving a file from a single server, we propose a parallel-access scheme where end users access multiple servers at the same time, fetching different portions of that file from different servers and reassembling them locally. The amount of data retrieved from a particular server depends on the resources available at that server or along the path from the user to the server. Faster servers will deliver bigger portions of a file while slower servers will deliver smaller portions. If the available resources at a server or along the path change during the download of a file, a dynamic parallel access will automatically shift the load from congested locations to less loaded parts (server and links) of the Internet. The end result is that users experience significant speedups and very consistent response times. Moreover, there is no need for complicated server selection algorithms and load is dynamically shared among all servers. The dynamic parallel-access scheme presented in this paper does not require any modifications to servers or content and can be easily included in browsers, peer-to-peer applications or content distribution networks to speed up delivery of popular content.**

***Index Terms*—Content distribution, HTTP, Internet, mirroring, parallel access, peer-to-peer, replication, Web.**

I. Introduction

IN ORDER to offload popular servers and improve end-user experience, copies of popular content are often stored in different locations. With network caching, geographically dispersed caches store copies of the documents required by their clients. With mirror site replication, documents from a primary site are proactively replicated at secondary sites. With peer-to-peer applications, users fetch and store content from other peer users in a effort to share the load among nodes at the edge of the network and bring content closer to the users.

When a copy of the same document exists at multiple servers, choosing the server that provides the best response time is not trivial and the resulting performance can dramatically vary depending on the server selected [6], [20]. Even when the fastest server has been selected, its performance can fluctuate during a download session, resulting sometimes in a poor response time at the end of the download. Rather than trying to choose the fastest available server, users can experience a better and more uniform performance by connecting to several servers that have an exact copy of the document. Instead of downloading the entire document from one server, a user downloads different parts of the same document from each of the servers in parallel. Once all the parts of the document are received, the user reconstructs the original document by reassembling the different parts.

In this paper, we propose a parallel-access scheme to download content from multiple servers at the same time. We consider two different parallel-access schemes: 1) **history-based TCP** parallel access, and 2) **dynamic TCP**. With a history-based parallel access, clients specify *a priori* which part of a document must be delivered from each mirror server, e.g., server one sends the first half of the document, and server two sends the second half. The portion of a document delivered by one server should be proportional to its service rate, thus, a slow server will deliver a small part of the document while a fast server will deliver a large part of the document. To calculate the portion of a document assigned to each server, a history-based parallel access uses a database of previous server rates, which is refreshed periodically, e.g., every few minutes. A history-based parallel-access scheme can speed up the download of a document when the network/server conditions are stable or easily predictable. However, when network/server conditions change rapidly, server rates are hard to predict and a history-based parallel access performs poorly.

With a dynamic parallel access, on the other hand, a client partitions a document into a large number of small blocks. The client first requests a different block from each server. Whenever a server finishes transmitting a block, the client issues a new request for a block that has not yet been requested from any other server. The same process is repeated for each block until all blocks are fully received. Note that at any given point in time all servers are kept busy sending a block (except for the idle times between block requests). When the client receives all blocks, it reassembles them to reconstruct the whole document.

There are several advantages to using a dynamic parallel access. First, since the block size is small, a dynamic parallel access can easily adapt to changing network/server conditions. The servers contacted share the load in a way that is proportional to the available resources at the server or in the path from the user to the server, therefore performing automatic load balancing. Fast servers will deliver larger portions of a document while slow servers will deliver smaller portions. This automatic load balancing is performed without any *a priori* information about server rates. Second, since the client is using

Manuscript received October 27, 2000; revised February 18, 2002; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Crovella. This work was supported in part by Eurecom's industrial partners, Ascom, Cegetel, France Telecom, Hitachi, Motorola, Swisscom, Texas Instruments Incorporated, and Thomson CSF.

P. Rodriguez is with Bell Laboratories Research, Holmdel, NJ 07733-3030 USA (e-mail: pablor@research.bell-labs.com).

E. W. Biersack is with the Institut Eurecom, 06904 Sophia Antipolis Cedex, France (e-mail: erbi@eurecom.fr).

Publisher Item Identifier 10.1109/TNET.2002.801413.

1063-6692/02$17.00 © 2002 IEEE

RED HAT 1012
RED HAT V. CAS

several connections to different servers, a parallel access is more resilient to congestion and failure in the network/servers than connecting to a single server. Load is automatically shifted from congested parts of the Internet to other parts with more abundant resources. Third, the server selection process is eliminated since clients connect to all available servers with a document copy. Fourth, the throughput seen by the client increases. Ideally, the total throughput seen by the client is equal to the sum of the bandwidths from each individual server to the client.

A parallel access, however, has some additional overhead compared a single access. There is an additional overhead incurred when opening multiple connections and extra traffic generated to perform block requests. (For a complete discussion on the costs of a parallel access, see Section VI-A.) To minimize the impact of these costs, a parallel access must be employed only with documents of a certain size, e.g., in the order of several hundreds of kilobytes. In the Web, the number of objects of this size is relatively small, thus, instead of suggesting a parallel access for general Web downloads, we propose a dynamic parallel access for other content types such as large documents, software downloads, music, video clips, or images.

Using analytical and experimental results, we evaluate the performance of a parallel-access scheme. We study the parallel-access behavior under different numbers of servers, different document sizes, and various network/server conditions including high-speed links as well as congested slow links. In addition, we use parallel-access implementation to better understand its advantages and limitations in a real deployment scenario and better define the scenarios where a parallel access is most beneficial.

The rest of the paper is organized as follows. Section II presents and analyzes a history-based parallel access. In Section III, we present the dynamic parallel access and demonstrate that it offers dramatic speedups for different document sizes, numbers of servers, and network conditions. Section IV considers a dynamic parallel access where a client is connected through a modem link. Section V compares a dynamic parallel access with a scheme where the client opens multiple parallel connections to the same server, and also studies the impact of request pipelining. Section VI discusses several important issues for the deployment of a parallel access. Section VII discusses related work and Section VIII concludes the paper.

II. History-Based Parallel Access

A history-based parallel access uses information about the previous transmission rates between the client and every mirror server. It needs this information to decide *a priori* which part of a document should be delivered by each server. The client divides a document into M disjoint blocks, and requests one block from every mirror server. Let μ_i be the transmission rate for server i, $1 \leq i \leq M$ and let S be the document size. If $\alpha_i S$ is the size of the block delivered by server i then $T_i = \alpha_i S/\mu_i$ denotes the download time of this block. To achieve a maximum speedup, all servers must finish transmitting their blocks at the same time, thus, $T_i = T_j$ for all $i, j \in \{1, \ldots, M\}$. When all servers transmit their blocks at the same time, there are no servers that stop transmitting before the document is fully received. The rate μ_p achieved with parallel access when all servers keep sending useful data until the document is fully received is equal to the sum of the individual rates of all servers, i.e., $\mu_p = \sum_{i=1}^{M} \mu_i$. Fast servers send a larger portion of the document, while slow servers send smaller portions. To achieve a maximum speedup, the size $\alpha_i S$ of the block sent by server i must be equal to $\alpha_i S = \mu_i/\mu_p$.

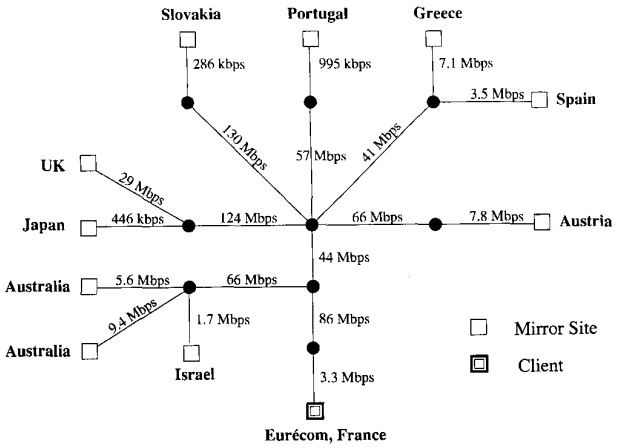


Fig. 1. Mirror servers for the Squid home page. Client is located at Eurecom, France.

A history-based parallel access needs to keep a database with information about the previous rates from the different servers to the receiver in order to estimate the rate μ_i to every server. Instead of having one database per client, a single database could be shared by a group of receivers connected through a proxy cache. The database is actualized every time a client connects to a server or can be updated periodically with an automated probing from the proxy.

A. Experimental Setup

To evaluate history-based parallel access, we have implemented this scheme as a Java client program that takes as input parameters the URLs and uses a database of previous rates from every mirror server to the client. The Java client performs a history-based parallel access for the requested document, saves the document locally, and records the time it took to download the document. To calculate the size of every block, clients need to know the total document size S. To obtain S, the parallel-access Java client polls the servers using an HTTP request at the beginning. The document size could also be prerecorded in a proxy cache or given to the client through a DNS server, thus avoiding additional round-trip time (RTT) to poll the servers.

To analyze the performance of a history-based parallel-access scheme, we performed several experiments using mirror servers in the Internet. In particular, we considered several mirror servers of the Squid Web page [22]. Fig. 1 shows a network map with the mirror servers considered and the bandwidth of the slowest link in every path as given by Pathchar [9]. The Java client is always located at Eurecom, France. Since the servers are situated in different countries and given that the connection

RED HAT 1012
RED HAT V. CAS

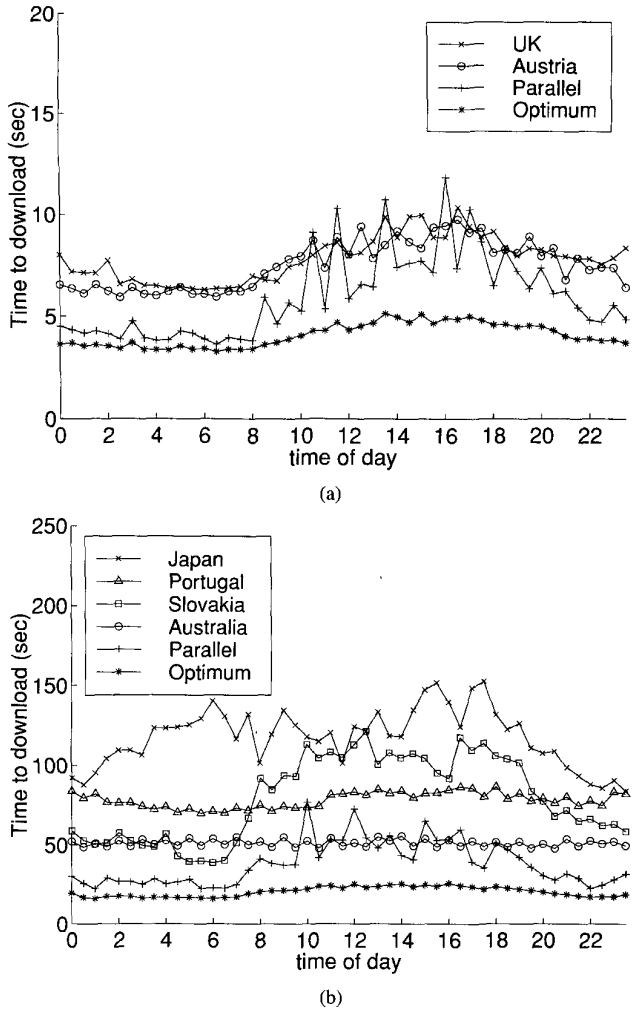


Fig. 2. History-based parallel access. (a) $S = 763$ kB, $M = 2$. (b) $S = 763$ kB, $M = 4$.

from our institution (Eurecom) into the Internet has a high access rate, a parallel-access connection from a Eurecom client to the mirror servers is likely to be bottleneck disjoint.

We evaluated a history-based parallel-access scheme every 15 minutes, making sure that different experiments did not overlap. We ran the experiments 24 hours a day during a ten-day period and averaged over the ten-day period.

B. Analysis of the Results

We present the performance results of a history-based parallel access where a client at Eurecom requests a document of 763 kB from two servers located in Austria and the U.K., which have average transmission rates between 80–100 kb/s. The document requested is the gzipped beta version of the Squid 1.2 software [22]. The database with the previous rates from the client to every server is updated when the Java client performs a request for the document, that is, every 15 minutes. The client assumes that the average rate μ_i offered by every server will be equal to the rate obtained 15 minutes before.

In Fig. 2, we show the download time obtained using a history-based parallel access and the download time obtained using an individual connection to every server. We see that during the night, when network conditions do not vary much, a history-based parallel access can efficiently estimate the average rate offered by every server and allows to significantly decrease the download time compared to the situation where the client accesses a single server. However, during the day, network conditions rapidly change and estimating the rate to every server by using the previously achieved rates results in poor estimates. Thus, the download times obtained with a history-based parallel access can be higher than the download times when clients access a single server. Similar performance for a history-based parallel access is also obtained for a different set of mirror servers [Fig. 2(b)]. In Fig. 2, we also show the **optimum** download time, which is defined as the download time achieved by a parallel-access scheme where all servers send useful information until the document is fully received. To calculate (*a posteriori*) the optimum download time, the average rates obtained from every server after the reception of a document are used.

To improve the performance history-based parallel-access schemes, the database could be refreshed more frequently during the day and other more sophisticated estimation algorithms could be used. However, finding the right refresh period and a good algorithm to estimate the rates is not an easy task. In Section III, we present another parallel-access scheme that does not require any past information and does not need to estimate the rates to the servers. Instead, the scheme dynamically adapts to the changing network conditions in real time.

III. Dynamic Parallel Access

We consider now a parallel-access scheme that uses dynamic requests between a client and the servers as the download of the document progresses. With a dynamic parallel access, the document is divided by the client into B blocks of **equal** size. In the case of Web access, a block is specified as a range of bytes in a document, e.g., from byte 100 to byte 200. A block requested can be specified using the HTTP 1.1 byte-range header [7]. The dynamic parallel-access scheme proceeds as follows.

- A client first requests one block from every server.
- Every time a client has completely received one block from a server, the client requests from this server another block that has not yet been requested from any other server.
- When the client has received all blocks, it reassembles them to reconstruct the whole document.

Since a client typically issues several requests to the same server during the download of a document, TCP-persistent connections are used between the client and every server to minimize the overhead of opening multiple TCP connections [17].

In order to best exploit the advantages of parallel access, one needs to keep all servers busy until all blocks have been received. In the dynamic parallel-access scheme, this is not the case.

- Each server will be idle between two consecutive block transfers (see Fig. 3). This idle time is referred to as *interblock* idle time and corresponds to one RTT. One can entirely avoid the interblock idle times by *pipelining*

RED HAT 1012
RED HAT V. CAS

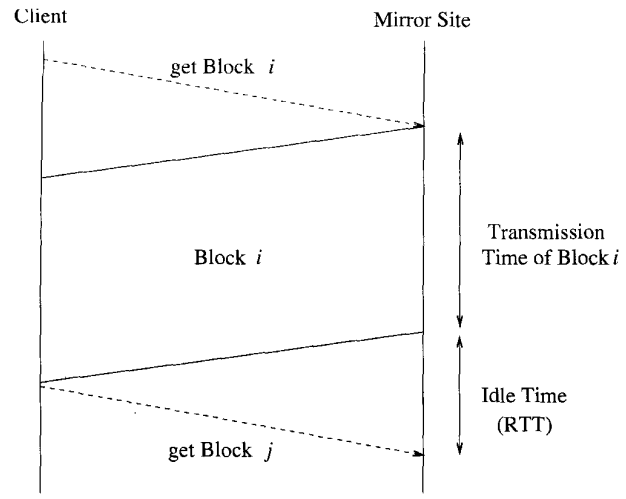


Fig. 3. Dynamic parallel access: Block request.

requests for different blocks to the same server. A request for a new block is made before the previous block is fully received, thus keeping the server continuously busy while there are still blocks to transmit. We will elaborate more on this point in Section VI-A.

- Not all servers terminate at the same time. If there are fewer than M blocks left (where M is the number of servers contacted) that have not been received, some servers will no longer transmit a block. The period of time since there are less than M servers transmitting blocks until the instant the document is fully received is referred to as *termination* idle time. The termination idle time is smaller or equal than $(S/B)/\mu_s$, where S/B is the block size and μ_s is the rate to the slowest server.

The following points need to be considered when determining the size of the blocks requested.

- The number B of blocks should be chosen to be much larger than the number M of mirror servers that are accessed in parallel.
- Each block should be small enough to provide a fine granularity of striping and ensure that the transfer of the last block requested from each server terminates at about the same time, thus fully utilizing the server and network resources until complete termination of the document transfer.
- Each block should also be sufficiently large as to keep the interblock idle time small compared to the download time of a block.

To reconcile the last two points, the document requested via parallel access must be sufficiently large, i.e., in the order of several hundreds of kilobytes.

A. Analysis of the Results

To evaluate the performance of dynamic parallel access, we implemented the scheme as a Java client. The Java client takes as input parameters the URLs of the servers with replicated content, performs a dynamic parallel access, saves the document locally, and records the transmission rate obtained. We evaluate the dynamic parallel-access scheme using the experimental setup described in Section II-A.

Our current implementation of dynamic parallel access does not consider pipelining to reduce the interblock idle times. However, the implementation reduces the termination idle time as follows. When there are fewer than M blocks missing, the client requests idle servers to deliver a block that is already requested from another server but that has not yet been fully received. With this approach, clients experience a transmission rate that is at least equal to the transmission rate of the fastest server at the expense of some bandwidth overhead. The bandwidth wasted in the worst case is equal to $\sum_{i=1}^{M-1} i(S/B)$, where S/B is the block size. The bandwidth wasted on average is much smaller than the worst case scenario since slow servers that did not complete the transmission of their last block are stopped after the document is fully received.

There are some more ways to minimize the bandwidth wasted.[1]

- The block size S/B of the blocks requested from slow servers can be decreased. More general, by dynamically adjusting the block size so that all servers finish at the same time, the bandwidth wasted would be zero. However, this approach requires accurate bandwidth estimations for each connection at runtime, which may be difficult to obtain [4].
- Slow servers can be stopped as soon as there are less than M blocks missing and request the missing bytes only of the last blocks from the fastest server.

We now compare the transfer time of dynamic parallel access with the transfer time of an individual access to every server and the optimum transfer time that would be achieved if there are neither intrablock nor termination idle times. The optimum transfer time provides a lower bound on the transfer time achievable by any implementation of a parallel-access scheme.

We first consider a dynamic parallel access to download a 763-kB document, which is replicated in $M = 4$ mirror servers (Fig. 4). The servers are located in Australia, Japan, Slovakia, and Portugal, to ensure disjoint paths. The average rate to these servers ranges from 5 to 15 kB/s, however, the instantaneous rates greatly fluctuate during the course of the day. We have chosen $B = 30$ blocks.

From Fig. 4 we can see that dynamic parallel access offers significant speedups compared to an individual document transfer from any single server. The transfer time is reduced from 50–150 to 20 s during all periods of the day. Even during highly congested periods, where the network conditions rapidly change, a dynamic parallel access offers very small transfer times. We also observe that the transfer time of a dynamic parallel access is very close to the optimum transfer time. Moreover, a dynamic parallel access that would implement pipelining to avoid interblock idle times would have performance almost equal to that of the optimum parallel scheme.

Next, we consider the situation with two fast servers (70 kB/s) and two slow ones (10 kB/s). The fast servers are

[1]These improvements have not been included in our implementation of the parallel-access scheme.

RED HAT 1012
RED HAT V. CAS

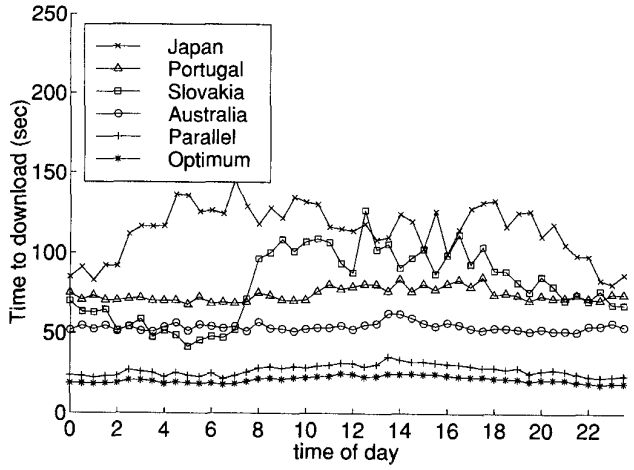


Fig. 4. Dynamic parallel access. $S = 763$ kB, $B = 30$, $M = 4$.

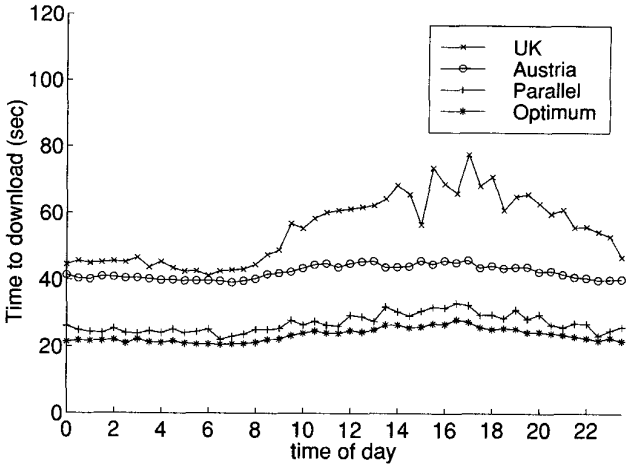


Fig. 6. Dynamic parallel access. $S = 5$ MB, $B = 40$, $M = 2$.

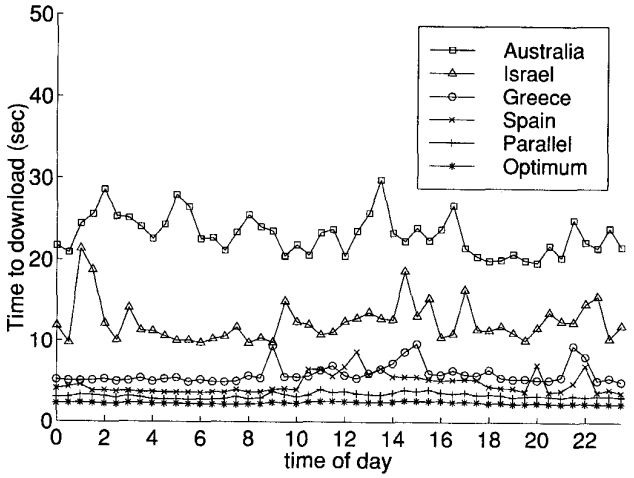


Fig. 5. Dynamic parallel access. $S = 256$ kB, $B = 20$, $M = 4$.

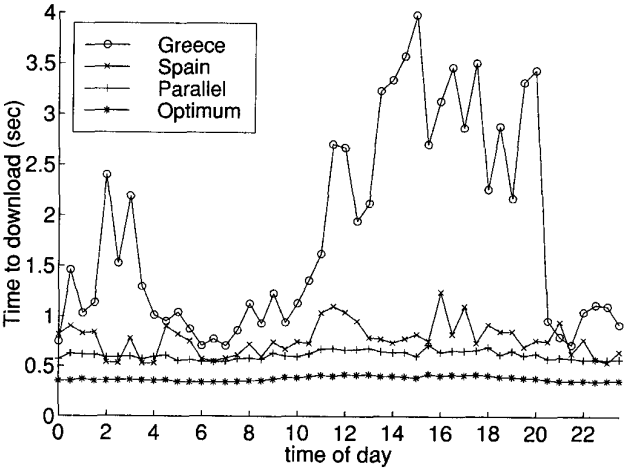


Fig. 7. Dynamic parallel access. $S = 10$ kB, $B = 4$, $M = 2$.

located in Greece and Spain, and the slow ones in Australia and Israel (Fig. 5). The document size is smaller than in the previous experiment, $S = 256$ kB, and we have also reduced the number of blocks to $B = 20$ to avoid that interblock idle times account for too high a percentage of the total transfer time (the document requested is the FAQ from Squid in Postscript format [22]).

We can see that a dynamic parallel-access scheme achieves a transfer time that is almost half the transfer time of the fast servers (slow servers contribute very few blocks and only slightly decrease the transfer time of the document). The latency benefits may not seem so important if they are compared to the case where a client connects to a fast server (from 4–5 to 2 s). However, if the client chooses the wrong server and connects to a slow server, it will end up experiencing transfer times up to 30 s.

In the next experiment, we consider only two mirror servers located in Austria and the U.K. and perform a dynamic parallel access for a large document of 5 MB (Fig. 6). Since both servers have a similar rate, a parallel access will reduce the transfer time by half. The time to download the document of 5 MB from a single server can be up to 80 s. Using a dynamic parallel access, the transfer rate is less than 30 s. The difference between the transfer time measured and that of the optimal parallel-access scheme is due to the interblock idle times.

B. Performance of Parallel Access With Small Documents

Even though a parallel-access scheme is not intended to be used with small documents, we study the performance of a dynamic parallel access with small documents of several kilobytes in size.

In Fig. 7, we see the performance of a dynamic parallel-access scheme for a 10-kB document. We consider two mirror servers located in Spain and Greece, and choose $B = 4$ blocks. We see that a dynamic parallel access has a download time that varies very little with the time of the day and is in most cases lower than the download time of the fastest server. Compared to the optimum download time, a dynamic parallel access has a much higher download time since the interblock idle times account for a high percentage of the total download time.

In addition, with small documents the connection setup time may account for a nonnegligible portion of the total download time. While a parallel-access scheme speeds up the download time of the document, it cannot affect the connection time. To

RED HAT 1012
RED HAT V. CAS

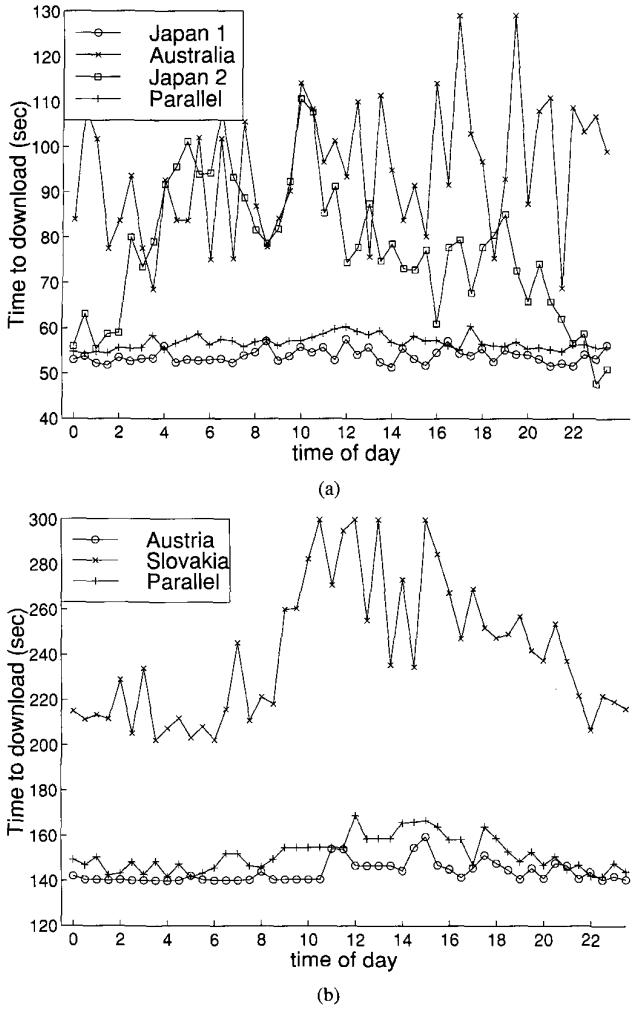


Fig. 8. Retrieval latency for a parallel-access scheme and for an individual-access scheme to every server over a shared bottleneck. (a) $S = 256$ kB, $B = 20$, $M = 3$. (b) $S = 763$ kB, $B = 30$, $M = 2$.

obtain better performances with a parallel access, several small documents could be grouped together, i.e., all documents in a Web page, to perform one dynamic parallel access to the larger document.

IV. Dynamic Parallel Access in the Case of a Shared Bottleneck Link

In this section, we study the performance of a dynamic parallel access where a client is connected through a modem link, i.e., a low-speed access link. In this case, the paths from the client to the servers are not bottleneck disjoint. A single server may already consume all the bandwidth of the modem link. Therefore, when a client uses another server in parallel, there is no residual network bandwidth and packets from different servers interfere and compete for bandwidth.

In Fig. 8, we consider dynamic parallel access for a client connected through a modem line at 56 kb/s. We show the download time achieved when connecting to every server individually and when connecting in parallel to all servers using a dynamic parallel access. In Fig. 8(a), we consider two slow servers (Japan 2 and Australia) and a fast server (Japan 1). In the case of an individual access to the fast server, the modem bandwidth is fully utilized and the download time varies little during all periods of the day. For an individual access to one of the slow servers, the rates obtained are about 24 kb/s, which is much lower than the modem speed of 56 kb/s. In this situation, the modem link is not fully utilized and the download time fluctuates depending on the different levels of congestion in the network/servers along the day. A similar effect can be seen in Fig. 8(b), where there are two mirror servers, one fast and one slow.

For the dynamic parallel access, we see that the download time achieved is close to the one of the fastest server, which is limited by the transmission rate of the modem link. The fact that the download time obtained with a dynamic parallel access is always slightly higher than the download time obtained for the fastest server is due to the interblock idle times. In Section IV-A, we study the performance of a dynamic parallel access that uses pipelining to avoid these idle times.

A. Evaluation of Request Pipelining

In this section, we repeat the previous experiments [Fig. 8(a) and (b)] and simulate a dynamic parallel access with block request pipelining. With request pipelining, a new block is requested from a server before the previously requested block is fully received. To fully avoid interblock idle times, a new block should be requested at least one RTT before the current block is completely received (see Fig. 3). Pipelining therefore requires a minimum block size. The block size should be such that $S/B > \text{RTT} \cdot \mu$. For instance, if the RTT between the client and most distant server is equal to RTT $= 100$ ms and the server has a transmission rate $\mu = 10$ kB/s, the block size must be $S/B > 1$ kB.

To estimate the improvement offered by request pipelining, we first measure for each server $i, i \in \{1, \ldots, M\}$ the download time tr_i obtained by parallel access without pipelining and the average round-trip time RTT_i. Then, we assume the interblock idle time to be equal to RTT_i. If server i has transmitted nb_i blocks, we estimate the download time tr_i^p with request pipelining as $tr_i^p = tr_i - (nb_i - 1) * \text{RTT}_i$. The estimated time T^p to completely download the document using pipelining is then given as $T^p = \max_{i \in \{1, \ldots, M\}} tr_i^p$.

In Fig. 9, we show the estimated download time T^p achieved by a parallel access with pipelining through a modem link. We observe that the download time of a parallel access with pipelining is smaller than [Fig. 9(a)] or equal to [Fig. 9(b)] the download time achieved by a single connection to the fastest server. In Fig. 9(a), the download time of each server is much smaller than the maximum modem link rate, thus, a single server connection does not fully utilize the modem link. In this case, a parallel access with pipelining can speed up the transfer of a document compared to a single server connection, and achieve download times that are even smaller than those offered by the fastest server. From Fig. 9(a) it is also important to notice that the transfer time achieved with a dynamic parallel access using pipelining is almost equal to the optimum download time. Thus, the additional delay that the Java implementation of dynamic parallel access introduces is very small.

RED HAT 1012
RED HAT V. CAS

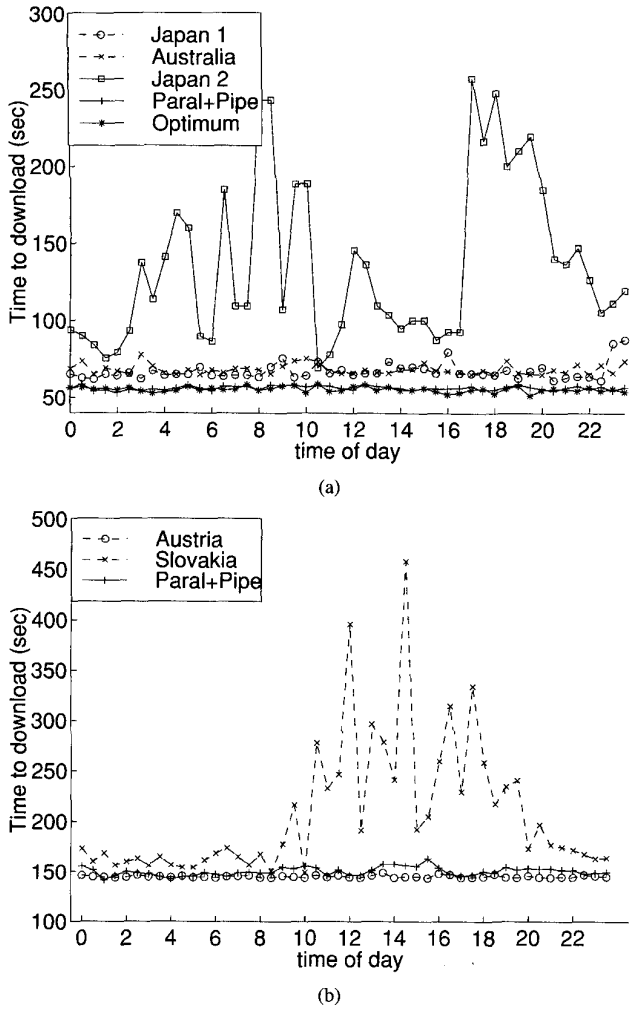


Fig. 9. Retrieval latency for a dynamic parallel-access scheme and an individual-access scheme to every server through a modem link, with pipelining. (a) $S = 256$ kB, $B = 20$, $M = 3$. (b) $S = 763$ kB, $B = 30$, $M = 2$.

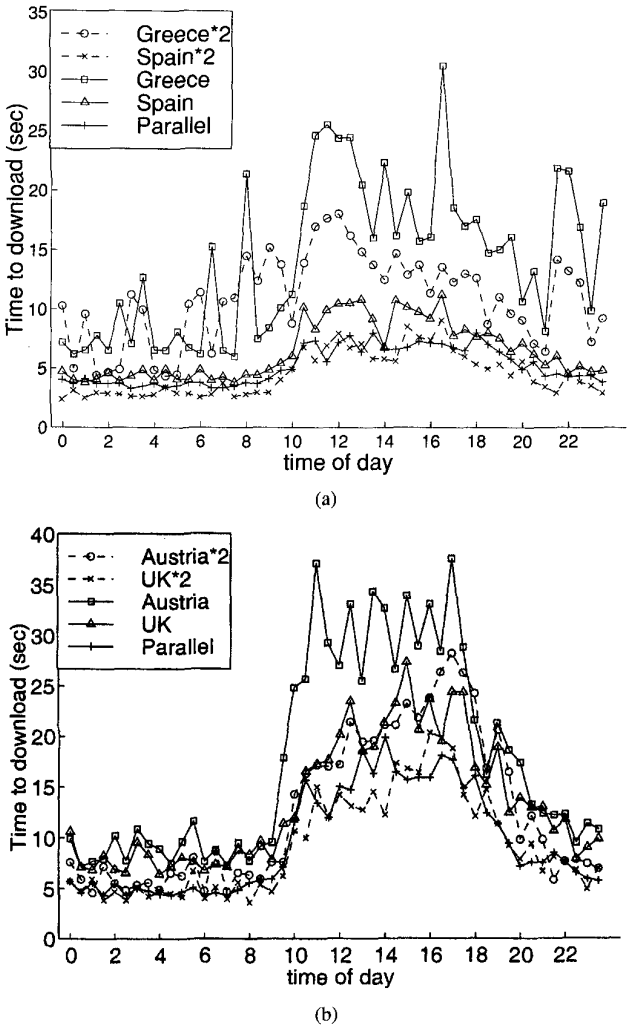


Fig. 10. Retrieval latency for a dynamic parallel-access scheme to $M = 2$ servers compared to a double parallel connection to the same server. (a) $S = 256$ kB, $B = 20$. (b) $S = 763$ kB, $B = 30$.

We would like to note that it is easy to implement pipelining. It does not require calculation of the exact RTTs but simply an estimate of an upper bound on the RTTs. Using an overestimated (too high) RTT, pipelining eliminates the idle times with no performance degradation.

V. Dynamic Parallel Access Versus Parallel Access to a Single Server

In this section, we compare a dynamic parallel access to multiple mirror servers with a parallel access to a *single* server. For a fair comparison, we consider the situation where a single client opens M TCP parallel connections to the same server and compare this case to dynamic parallel access to M servers. Let μ_s be the rate to the slowest server, and μ_f be the rate to the fastest server. If the residual bandwidth of the path from the client to the server is large enough, an M parallel connection to a single server with rate μ_i will have in the best case a transmission rate equal to $M \cdot \mu_i$. A dynamic parallel access to M servers has a transmission rate $\mu_p = \sum_{i=1}^{M} \mu_i$, which is higher than the transmission rate of an M parallel access to the slowest server, but smaller than the transmission rate of an M parallel access to the fastest server, $M \cdot \mu_s \leq \mu_p \leq M \cdot \mu_f$.

Next, we consider the situation where there are two mirror servers, a slow one in Greece and a fast one in Spain, and perform the following experiments: 1) clients retrieve a document using a single connection to each server; 2) clients retrieve a document using a dynamic parallel access to both servers; 3) clients retrieve a document using a dynamic parallel access with two connections to the same server.

Fig. 10 shows the download time obtained for the different schemes and for two different document sizes. For the fast server in Spain, the available resources from the client to the server are abundant, and, therefore, two parallel connections to this server result in a reduction of the download time compared to a single connection to the same server. However, when two connections are simultaneously opened to the slow server in Greece, the resulting download time is sometimes higher than the download time obtained if the client opened only one connection to this server. This is due to the fact that the server or network path to Greece is very instable and variable. Thus, opening two TCP

RED HAT 1012
RED HAT V. CAS

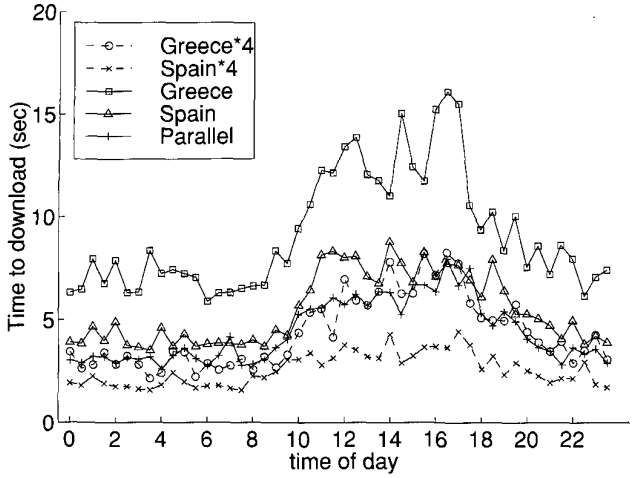


Fig. 11. Retrieval latency for a dynamic parallel-access scheme to $M = 2$ servers compared to a four parallel connections to the same server. $S = 256$ kB, $B = 20$.

connections to the same server does not ensure better response times, because the server or the networks may be experiencing a high load at that time. With a dynamic parallel access to both servers, on the other hand, the load dynamically shifts to use resources where they are available, thus keeping the load on congested servers/network paths low. As a result, the download time for a dynamic parallel access to both servers is smaller than a parallel access to the slowest server and quite comparable to a parallel access to the fastest server only.

In Fig. 11, we have considered the situation where the client opens four parallel connections to a single server and we compare the obtained speedup with that of a dynamic parallel access to both servers. We can see that for both servers in Greece and Spain, opening four connections to the same server gives better performance than opening just one. While opening four connections to the fast server in Spain offers smaller download times than a dynamic parallel access to both servers, in the case of opening four connections to the slow server in Greece, the download time is equal to or even higher than a dynamic parallel access to both servers.

Therefore, while parallel connections to a single server may result in high speedups if the fastest server is selected, they will also result in little speedup if a slow server is selected, even for a high number of concurrent connections. On the other hand, a dynamic parallel access to both servers automatically achieves very good speedup without any server selection. Moreover, when using multiple connections to the same server, the links close to the server or the actual server may become congested, and clients will not experience any speedup. With a dynamic parallel access to different servers, on the other hand, the load is shared among the servers and a higher number of receivers can experience significant speedups.

VI. Discussion

A. Cost of Parallel Access

A parallel access improves the download performance and provides several other performance advantages compared to accessing a single server. However, a parallel access also has several costs involved that need to be considered.

There is the overhead of the extra server access to find out the document size (this could be done with a HEAD request [7]). Note that for large documents the overhead incurred in obtaining the document size is negligible. Moreover, issuing a first block request with a predetermined fixed size, which would include the document size plus some data, and then using a parallel access on further block requests, could eliminate this overhead.

There is the overhead incurred by the block request messages. To reduce this cost, an intelligent block assignment policy can be used to gradually increase the size of those blocks assigned to fast servers.

There is an increase in the overall number of TCP connections compared to a single server access. However, the duration of the TCP connections with a parallel access is smaller since each server delivers only a portion of the whole document. Also, the connection setup is relatively insignificant when large documents are considered. Finally, using servers in different time zones (e.g., selecting servers from places in the world where it is night) reduces the impact of opening multiple connections since idle servers can be selected.

B. When to Use a Parallel Access: Benefits and Limitations

To outweigh the costs described in the previous section, a parallel access must be applied to large files in the order of several hundreds of kilobytes. Examples of such files are large document downloads, software packages, music, video, and large images. In the Web, most of the files tend to be quite small, and usually only the root file of a Web page is large enough to benefit from a parallel access. Nevertheless, parallel access can be used to download Web content if several Web objects are bundled together into a large file. This technique allows, for instance, to rapidly prefetch all the components of a Web page into a Web proxy cache.

When clients access the Internet through slow links, e.g., a modem link, the rate of each server may be higher than the rate at the client's access link. In this case, performing a parallel access may result in no additional speedup versus an individual access to one server. However, the server rates are usually not known *a priori* by the client and servers or network conditions may fluctuate during a download session. A parallel access reduces the uncertainty of selecting a very slow or instable server and brings the client's access link to its full utilization, thus providing a faster and more consistent experience. One important point to note in this scenario is that disjoint paths are not a necessary condition for speedup. Instead, a necessary condition is spare or abundant resources (both at the server and along the path).

When clients access the Internet through fast links, it is very likely that the bottleneck capacity is somewhere in the network or in the servers themselves. As more and more users access the same popular content, the likelihood of sharing the same bottleneck rapidly increases. When the bottleneck resources are fully utilized, all users compete for the same resources and increasing the number of parallel-access connections per user does not result in additional speedups. Though the benefits of a parallel access in a shared bottleneck environment are limited, a parallel

RED HAT 1012
RED HAT V. CAS

access still provides a much better experience than selecting a single server since it avoids the risk of selecting a very slow server. Moreover, a parallel access provides the same average performance to *all* users that share the same bottleneck since it prevents that some users are assigned to fast servers and experience low download times, while other users are assigned to very slow servers and experience high download times. Besides, a parallel access hides sudden changes in server/link performance by shifting load from servers/links that are overloaded to other parts of the Internet where resources are available.

One way to reduce the probability that all users share the same bottleneck and thus improve the speedup offered by a parallel access is to use a dynamic parallel access in the context of peer-to-peer applications or content-distribution networks. In a peer-to-peer environment, when a client downloads content from another peer client, the new client becomes itself a server for future clients. As content becomes more and more popular, it gets replicated in new mirror sites throughout the network. These new mirror sites can be used as new sources to perform multiple parallel accesses that avoid competition for the same resources.

In summary, we see the greatest potential for dynamic parallel access in the area of peer-to-peer applications such as Napster,[2] where the size of content accessed is typically large and the machines storing these copies of the same content are distributed over a large geographical area, which helps improve path disjointness. An application such as Napster also has the advantageous property that the more popular the content, the more mirror servers there are. This is due to the fact that as content becomes more and more popular, i.e., an increasing number of clients have been downloading this content, all these clients potentially become mirror servers.

C. Deployment Issues

These are a number of possible issues that should be considered when deploying a parallel-access system.

Concerning the discovery of mirror servers, the most frequent approach is to publish a list of mirror servers on the master Web server. Clients manually select the server that they believe will offer the fastest retrieval time. Some search engines, e.g., Go!zilla,[3] provide a full list of mirror servers and rate them in terms of loss rate and RTT. Several organizations which run mirror servers have modified DNS servers or delegate the DNS resolution to other modified DNS servers that return to the client the IP addresses of the administratively closest mirror servers/caches [1].[4] Other recent studies suggest extending DNS servers [10] or creating a central directory [8] to return a full list of all servers containing a copy of a certain document. Current cache-sharing protocols [19], [5] keep local information about the location of duplicated document copies in neighbor caches. When a client requests a certain document and the document is not found in the local cache, the local cache will redirect the request to the best neighbor cache with a document copy.

To start a parallel access, the client needs to know the document size. The document size could be given to the clients when they obtain the list of servers (from a central repository, a modified DNS, the origin server, etc.). If this is not the case, the client can obtain the document size with a first initial request for a fixed small block.

The number of servers that a client connects to does not necessarily need to comprise all mirror servers. In fact, most of the performance improvement offered by a parallel access can be achieved using a few servers [16], [11]. Instead of using all mirror servers, clients can obtain a reduced subset of mirror servers and perform a parallel access in this subset. Obtaining a subset of servers requires some server selection process, however, this scoping process is still much easier than selecting the fastest server.

In the case where there are several slow servers and a few mirror servers with relatively high transmission rates, almost all blocks will be requested from these fast servers. In this case, a parallel access may wish to drop the connections from the slow servers to conserve server resources without significantly affecting performance. Moreover, if a server happens to be fast enough to use all the bandwidth in the access link of the end user, a parallel access may decide to continue only with this fast server or continue using multiple servers for load-balancing reasons and to improve reliability and resilience against possible congestion.

Parallel access to Web documents requires that clients and servers support range requests as specified in HTTP 1.1 and TCP-persistent connections. However, current server implementations of range requests and TCP-persistent connections have some unpredictable behavior that can affect the performance of a parallel access. For instance, servers are not required to honor every range request and there may be cases where servers respond with data that does not cover the requested range. Also, servers may choose not to honor range requests for certain types of files. Moreover, not all servers support persistent connections and may decide to close the connection after each request transaction. For a more detailed study of the peculiarities of HTTP 1.1 server implementations, see [12]. One way to solve these problems is by placing a reverse proxy in front of a server or farm of servers that implements persistent connections as well as range requests.

Parallel access only works when the replicated content is identically replicated. To make sure that different servers store exactly the same version of a document, servers could provide a hashing tag that determines the current version of the document.

Parallel access can be deployed in client browsers, cache sharing protocols, or content distribution networks. Moreover, parallel access can be easily integrated in peer-to-peer schemes such as Napster and Gnutella.[5]

VII. Related Work

Currently there exist several software packages that allow clients to dynamically pause, resume, and jump from one mirror server to another during a document download if the current

[2]Napster. [Online.] Available: http://www.napster.com.

[3]Go!zilla. [Online.] Available: http://www.gizmo.net/.

[4]World Wide Web Consortium. [Online.] Available: http://www.w3c.org.

[5]Gnutella. [Online.] Available: http://gnutella.wego.com.

RED HAT 1012
RED HAT V. CAS

mirror server is very slow, including Go!zilla, Netscape Smart Download,[6] and Leechftp.[7] Other software packages, such as Agiletp,[8] allow to open multiple parallel connections to a certain server to speed the download of a certain document.

Choosing the best mirror server has been the subject of research during recent years. Several techniques have been proposed, including multicast communication to poll all the mirror servers [2], dynamical probing, combining server push with client probes[6], and statistical recordkeeping [21]. The work in [21] indicates that the choice of the best server is often not obvious and that the obtained performance can dramatically vary depending on the server selected.

Maxemchuk's work on dispersity routing [15] and Rabin's work on information dispersal [18] explored how to improve document delivery from a *single* server along *multiple paths*. Using erasure codes, the server takes the original document, breaks it into k blocks, and generates h parity blocks with the property that *any* k out of the $k+h$ data/parity blocks can be used to reconstruct the original k blocks. By transmitting more than k blocks, the server reduces the download time and increases the probability that the receiver is able to reconstruct the original document even if some of the blocks are lost.

Byers *et al.* [3] proposed to access *multiple* servers in parallel. They used an open-loop multicast distribution where different servers generate different sets of parity blocks and cyclically transmit parities and originals. Clients can recover the whole document as soon as they receive enough (k) different blocks, regardless of which servers the blocks came from [3]. To efficiently encode large documents with small encoding/decoding delays, special erasure codes [13], such as Tornado Codes [14], must be used. However, this approach has a major drawback compared to the dynamic parallel-access scheme discussed in this paper. It requires the servers to *encode all their documents* and the clients to *install decoders* to reconstruct the encoded documents. In addition, some problems still remain unresolved, such as how to stop the servers or how to perform congestion control.

For these reasons, we propose a parallel-access scheme where clients and servers communicate via unicast using TCP. To the best of our knowledge. our implementation of dynamic parallel access is the first parallel-access system that uses standard TCP and HTTP protocols to dynamically request different pieces of a document from the mirror servers, and does not require re-encoding of the content.

More recent experiments based on the parallel-access technique described in this paper were presented in [16]. Several tests were performed for a different implementations of dynamic parallel access in an environment where the available bandwidth between the client and the servers is much higher (a factor of 3 to 7) than in our experiments. The results obtained indicate that the performance of a parallel access in such scenarios can be smaller than the performance results presented in this paper. However, the implementation in [16] can benefit from a series of optimizations used in our paper to significantly improve the performance. The main optimization consists in keeping all servers busy all the time by avoiding the interblock and termination idle times. Using these enhancements, a parallel access continues to offer significant performance improvements compared to a single server selection even in a high-bandwidth scenario for documents of several hundreds of kilobytes.

Our implementation of dynamic parallel access can be easily integrated in Web browsers without any modification of the mirror servers and no additional buffer requirements at the clients (since current Web browsers already support opening multiple parallel connections to the same server). It can also be included in cache sharing protocols [5], [19] to share the load among neighbor caches with the same document copy or content-distribution networks [1] to speed up the download of popular documents into proxy caches.

The dynamic parallel-access scheme described in this paper has already been integrated into several peer-to-peer applications such as Morpheus[9] and OpenCola,[10] which are used to download software, music, images, or video. Both applications provide a mechanism to identify all the peer servers that store the same copy of the content, and they perform a parallel access to speed up content delivery. More recently, companies such as Kontiki[11] are using a content-distribution network based on end hosts to provide efficient content delivery. Kontiki uses a dynamic parallel-access scheme to eliminate the selection process while speeding up content delivery and offering a more consistent user experience.

VIII. Conclusion and Future Work

Given that popular documents are often replicated on multiple servers, we suggested that clients connect in parallel to several mirror servers for retrieving a document. We presented a dynamic parallel-access scheme that speeds up document downloads, automatically balances the load among servers, and avoids complex server selection.

We implemented a dynamic parallel-access scheme and evaluated its performance for different numbers of servers, document sizes, and network conditions. We showed that dynamic parallel access achieves significant speedups and has a performance that comes very close to the optimum performance that can be achieved with any parallel-access scheme. Even when clients are connected through modem lines, a dynamic parallel access offers a download time that is close to the download time of the fastest server without any server selection. A dynamic parallel-access scheme can be easily implemented and does not require modifications of the content in the mirror servers, in contrast with the digital fountain approach which requires re-encoding of the document [3].

Future versions of our implementation will include pipelining of several blocks to avoid idle times. However, the expected improvement will be modest since a dynamic parallel access without pipelining already gives download times that are very close to the optimum ones. To reduce the number of negotiations

[6]Netscape Smart Download. [Online.] Available: http://www.netscape.com/.

[7]Leechftp. [Online.] Available: http://linux.fh-heilbronn.de/debis/leechftp/.

[8]Agiletp. [Online.] Available: http://www.daemon-info.com/Us/products.htm.

[9]MusicCity. [Online.] Available: http://www.musiccity.com.

[10]OpenCola Swarmcast. [Online.] Available: http://www.opencola.org/projects/swarmcast.shtml.

[11]Kontiki. [Online.] Available: http://www.kontiki.com.

RED HAT 1012
RED HAT V. CAS

between the client and the servers, clients could keep track of the fastest server during the download of the first blocks and, instead of using a fixed block size, dynamically increase the block size for the fast servers. This approach would require more complexity at the client, but seems a natural extension to our scheme.

The integration of dynamic parallel access into several existing peer-to-peer applications and content-distribution networks is an indication of the importance of dynamic parallel access for speeding up object download. Audio/video content is particularly well suited for parallel access since it is usually static, large, and popular. Easy extensions of the parallel-access technique described in this paper could also be used to greatly improve audio/video streaming performance.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments.

REFERENCES

[1] *FreeFlow: How it Works*, Akamai Technologies, Cambridge, MA, Nov. 1999.

[2] J. Bernabeu, M. Ammar, and M. Ahamad, "Optimizing a generalized polling protocol for resource finding over a multiple access channel," *Comput. Netw. ISDN Syst.*, vol. 27, pp. 1429–1445, 1995.

[3] J. Byers, M. Luby, and M. Mitzenmacher, "Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads," in *Proc. IEEE INFOCOM*, vol. 1, New York, NY, Mar. 1999, pp. 275–283.

[4] R. L. Carter and M. E. Crovella, "Server selection using dynamic path characterization in wide-area networks," in *Proc. IEEE INFOCOM*, vol. 3, Kobe, Japan, Apr. 1997, pp. 1014–1021.

[5] L. Fan, P. Cao, J. Almeida, and A. Broder, "Summary cache: A scalable wide-area web cache sharing protocol," in *Proc. ACM SIGCOMM*, Feb. 1998, pp. 254–265.

[6] Z. Fei, S. Bhattacharjee, E. W. Zegura, and M. H. Ammar, "A novel server selection technique for improving the response time of a replicated service," in *Proc. IEEE INFOCOM*, vol. 2, San Francisco, CA, Mar. 1998, pp. 783–791.

[7] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee, "Hypertext transfer protocol—HTTP/1.1," Network Working Group, RFC 2068, Jan. 1997.

[8] S. Gadde, M. Rabinovich, and J. Chase, "Reduce, reuse, recycle: An approach to building large Internet caches," in *Proc. 6th Workshop Hot Topics in Operating Systems (HotOS-VI)*, Cape Cod, MA, May 1997, pp. 93–98.

[9] V. Jacobson. Pathchar. Lawrence Berkeley Lab Network Research Group. [Online]. Available: http://www.caida.org/tools/utilities/others/pathchar/. Source: ftp://ftp.ee.lbl.gov/pathchar.

[10] J. Kangasharju, K. W. Ross, and J. Roberts, "Locating copies of objects using the domain name system," presented at the 4th Int. Web Caching Workshop, San Diego, CA, Mar. 1999.

[11] A. Kirpal, "Study of parallel access schemes to speed up the internet," M.S. thesis, Univ. Munich, Munich, Germany/Institut Eurécom, Sophia Antipolis, France, Apr. 1999.

[12] B. Krishnamurthy, J. Mogul, and D. Kirstol, "Key differences between HTTP/1.0 and HTTP/1.1," presented at the 8th Int. World Wide Web Conf., Toronto, ON, Canada, May 1999.

[13] M. Luby, "Information additive code generator and decoder for communication systems," U.S. Patent 6 307 487, Oct. 23, 2001.

[14] M. Luby , M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. ACM Symp. Theory of Computing (STOC)*, El Paso, TX, May 1997, pp. 150–159.

[15] N. F. Maxemchuk, "Dispersity routing in store-and-forward networks," Ph.D. dissertation, Univ. Pennsylvania, Philadelphia, 1975.

[16] A. Miu and E. Shih, "Performance analysis of a dynamic parallel downloading scheme from mirror sites throughout the Internet," Term paper, MIT Lab. Comput. Sci., Cambridge, MA, Dec. 1999.

[17] V. N. Padmanabhan and J. Mogul, "Improving HTTP latency," in *Proc. 2nd World Wide Web Conf. Mosaic and the Web*, Chicago, IL, Oct. 1994, pp. 995–1005.

[18] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *J. Assoc. Comput. Mach.*, vol. 36, no. 2, pp. 335–348, Apr. 1989.

[19] A. Rousskov and D. Wessels, "Cache digests," *Comput. Networks ISDN Syst.*, vol. 30, no. 22–23, pp. 2155–2168, 1998.

[20] M. Sayal, Y. Breibart, P. Scheuermann, and R. Vingralek, "Selection algorithm for replicated web servers," presented at the Workshop Internet Server Performance (WISP/SIGMETRICS), Madison, WI, June 1998.

[21] S. Sesham, M. Stemm, and R. Katz, "SPAND: Shared passive network performance discovery," presented at the USENIX Symp. Internet Technologies and Systems, Monterey, CA, Dec. 1997.

[22] D. Wessels. (1996) Squid internet object cache. [Online]. Available: http://www.nlanr.net/Squid/

Pablo Rodriguez received the M.S. degree in telecommunication engineering from the University of Navarra (UPNA), Pamplona, Spain, and the Ph.D. degree from the Institut Eurecom, Sophia Antipolis, France.

He is currently a Member of Technical Staff with Bell Laboratories Research, Holmdel, NJ.

Ernst W. Biersack (M'88) received the M.S. and Ph.D. degrees in computer science from the Technische Universität München, Munich, Germany.

From March 1989 to February 1992, he was a Member of Technical Staff with the Computer Communications Research District, Bell Communications Research, Morristown NJ. Since March 1992, he has been a Professor of Telecommunications at the Institut Eurecom, Sophia Antipolis, France.

RED HAT 1012
RED HAT V. CAS

Network Working Group
Request for Comments: 2663
Category: Informational

P. Srisuresh
M. Holdrege
Lucent Technologies

August 1999

IP Network Address Translator (NAT) Terminology and Considerations

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Preface

The motivation behind this document is to provide clarity to the terms used in conjunction with Network Address Translators. The term "Network Address Translator" means different things in different contexts. The intent of this document is to define the various flavors of NAT and standardize the meaning of terms used.

The authors listed are editors for this document and owe the content to contributions from members of the working group. Large chunks of the document titled, "IP Network Address Translator (NAT)" were extracted almost as is, to form the initial basis for this document. The editors would like to thank the authors Pyda Srisuresh and Kjeld Egevang for the same. The editors would like to thank Praveen Akkiraju for his contributions in describing NAT deployment scenarios. The editors would also like to thank the IESG members Scott Bradner, Vern Paxson and Thomas Narten for their detailed review of the document and adding clarity to the text.

Abstract

Network Address Translation is a method by which IP addresses are mapped from one realm to another, in an attempt to provide transparent routing to hosts. Traditionally, NAT devices are used to connect an isolated address realm with private unregistered addresses to an external realm with globally unique registered addresses. This document attempts to describe the operation of NAT devices and the associated considerations in general, and to define the terminology used to identify various flavors of NAT.

RED HAT 1012
582
RED HAT V. CAS

1. Introduction and Overview

The need for IP Address translation arises when a network's internal IP addresses cannot be used outside the network either because they are invalid for use outside, or because the internal addressing must be kept private from the external network.

Address translation allows (in many cases, except as noted in sections 8 and 9) hosts in a private network to transparently communicate with destinations on an external network and vice versa. There are a variety of flavors of NAT and terms to match them. This document attempts to define the terminology used and to identify various flavors of NAT. The document also attempts to describe other considerations applicable to NAT devices in general.

Note, however, this document is not intended to describe the operations of individual NAT variations or the applicability of NAT devices.

NAT devices attempt to provide a transparent routing solution to end hosts trying to communicate from disparate address realms. This is achieved by modifying end node addresses en-route and maintaining state for these updates so that datagrams pertaining to a session are routed to the right end-node in either realm. This solution only works when the applications do not use the IP addresses as part of the protocol itself. For example, identifying endpoints using DNS names rather than addresses makes applications less dependent of the actual addresses that NAT chooses and avoids the need to also translate payload contents when NAT changes an IP address.

The NAT function cannot by itself support all applications transparently and often must co-exist with application level gateways (ALGs) for this reason. People looking to deploy NAT based solutions need to determine their application requirements first and assess the NAT extensions (i.e., ALGs) necessary to provide application transparency for their environment.

IPsec techniques which are intended to preserve the Endpoint addresses of an IP packet will not work with NAT enroute for most applications in practice. Techniques such as AH and ESP protect the contents of the IP headers (including the source and destination addresses) from modification. Yet, NAT's fundamental role is to alter the addresses in the IP header of a packet.

2. Terminology and concepts used

Terms most frequently used in the context of NAT are defined here for reference.

RED HAT 1012
583
RED HAT V. CAS

2.1. Address realm or realm

An address realm is a network domain in which the network addresses are uniquely assigned to entities such that datagrams can be routed to them. Routing protocols used within the network domain are responsible for finding routes to entities given their network addresses. Note that this document is limited to describing NAT in IPv4 environment and does not address the use of NAT in other types of environment. (e.g. IPv6 environments)

2.2. Transparent routing

The term "transparent routing" is used throughout the document to identify the routing functionality that a NAT device provides. This is different from the routing functionality provided by a traditional router device in that a traditional router routes packets within a single address realm.

Transparent routing refers to routing a datagram between disparate address realms, by modifying address contents in the IP header to be valid in the address realm into which the datagram is routed. Section 3.2 has a detailed description of transparent routing.

2.3. Session flow vs. Packet flow

Connection or session flows are different from packet flows. A session flow indicates the direction in which the session was initiated with reference to a network interface. Packet flow is the direction in which the packet has traveled with reference to a network interface. Take for example, an outbound telnet session. The telnet session consists of packet flows in both inbound and outbound directions. Outbound telnet packets carry terminal keystrokes and inbound telnet packets carry screen displays from the telnet server.

For purposes of discussion in this document, a session is defined as the set of traffic that is managed as a unit for translation. TCP/UDP sessions are uniquely identified by the tuple of (source IP address, source TCP/UDP port, target IP address, target TCP/UDP port). ICMP query sessions are identified by the tuple of (source IP address, ICMP query ID, target IP address). All other sessions are characterized by the tuple of (source IP address, target IP address, IP protocol).

Address translations performed by NAT are session based and would include translation of incoming as well as outgoing packets belonging to that session. Session direction is identified by the direction of the first packet of that session (see sec 2.5).

RED HAT 1012
RED HAT V. CAS

Note, there is no guarantee that the idea of a session, determined as above by NAT, will coincide with the application's idea of a session. An application might view a bundle of sessions (as viewed by NAT) as a single session and might not even view its communication with its peers as a session. Not all applications are guaranteed to work across realms, even with an ALG (defined below in section 2.9) enroute.

2.4. TU ports, Server ports, Client ports

For the reminder of this document, we will refer TCP/UDP ports associated with an IP address simply as "TU ports".

For most TCP/IP hosts, TU port range 0-1023 is used by servers listening for incoming connections. Clients trying to initiate a connection typically select a source TU port in the range of 1024-65535. However, this convention is not universal and not always followed. Some client stations initiate connections using a source TU port number in the range of 0-1023, and there are servers listening on TU port numbers in the range of 1024-65535.

A list of assigned TU port services may be found in RFC 1700 [Ref 2].

2.5. Start of session for TCP, UDP and others

The first packet of every TCP session tries to establish a session and contains connection startup information. The first packet of a TCP session may be recognized by the presence of SYN bit and absence of ACK bit in the TCP flags. All TCP packets, with the exception of the first packet, must have the ACK bit set.

However, there is no deterministic way of recognizing the start of a UDP based session or any non-TCP session. A heuristic approach would be to assume the first packet with hitherto non-existent session parameters (as defined in section 2.3) as constituting the start of new session.

2.6. End of session for TCP, UDP and others

The end of a TCP session is detected when FIN is acknowledged by both halves of the session or when either half receives a segment with the RST bit in TCP flags field. However, because it is impossible for a NAT device to know whether the packets it sees will actually be delivered to the destination (they may be dropped between the NAT device and the destination), the NAT device cannot safely assume that the segments containing FINs or SYNs will be the last packets of the session (i.e., there could be retransmissions). Consequently, a session can be assumed to have been terminated only after a period of

RED HAT 1012
585
RED HAT V. CAS

4 minutes subsequent to this detection. The need for this extended wait period is described in RFC 793 [Ref 7], which suggests a TIME-WAIT duration of 2 * MSL (Maximum Segment Lifetime) or 4 minutes.

Note that it is also possible for a TCP connection to terminate without the NAT device becoming aware of the event (e.g., in the case where one or both peers reboot). Consequently, garbage collection is necessary on NAT devices to clean up unused state about TCP sessions that no longer exist. However, it is not possible in the general case to distinguish between connections that have been idle for an extended period of time from those that no longer exist. In the case of UDP-based sessions, there is no single way to determine when a session ends, since UDP-based protocols are application specific.

Many heuristic approaches are used to terminate sessions. You can make the assumption that TCP sessions that have not been used for say, 24 hours, and non-TCP sessions that have not been used for a couple of minutes, are terminated. Often this assumption works, but sometimes it doesn't. These idle period session timeouts vary a great deal both from application to application and for different sessions of the same application. Consequently, session timeouts must be configurable. Even so, there is no guarantee that a satisfactory value can be found. Further, as stated in section 2.3, there is no guarantee that NAT's view of session termination will coincide with that of the application.

Another way to handle session terminations is to timestamp entries and keep them as long as possible and retire the longest idle session when it becomes necessary.

2.7. Public/Global/External network

A Global or Public Network is an address realm with unique network addresses assigned by Internet Assigned Numbers Authority (IANA) or an equivalent address registry. This network is also referred as External network during NAT discussions.

2.8. Private/Local network

A private network is an address realm independent of external network addresses. Private network may also be referred alternately as Local Network. Transparent routing between hosts in private realm and external realm is facilitated by a NAT router.

RFC 1918 [Ref 1] has recommendations on address space allocation for private networks. Internet Assigned Numbers Authority (IANA) has three blocks of IP address space, namely 10/8, 172.16/12, and 192.168/16 set aside for private internets. In pre-CIDR notation, the

RED HAT 1012
RED HAT V. CAS

first block is nothing but a single class A network number, while the second block is a set of 16 contiguous class B networks, and the third block is a set of 256 contiguous class C networks.

An organization that decides to use IP addresses in the address space defined above can do so without coordination with IANA or any other Internet registry such as APNIC, RIPE and ARIN. The address space can thus be used privately by many independent organizations at the same time. However, if those independent organizations later decide they wish to communicate with each other or the public Internet, they will either have to renumber their networks or enable NAT on their border routers.

2.9. Application Level gateway (ALG)

Not all applications lend themselves easily to translation by NAT devices; especially those that include IP addresses and TCP/UDP ports in the payload. Application Level Gateways (ALGs) are application specific translation agents that allow an application on a host in one address realm to connect to its counterpart running on a host in different realm transparently. An ALG may interact with NAT to set up state, use NAT state information, modify application specific payload and perform whatever else is necessary to get the application running across disparate address realms.

ALGs may not always utilize NAT state information. They may glean application payload and simply notify NAT to add additional state information in some cases. ALGs are similar to Proxies, in that, both ALGs and proxies facilitate Application specific communication between clients and servers. Proxies use a special protocol to communicate with proxy clients and relay client data to servers and vice versa. Unlike Proxies, ALGs do not use a special protocol to communicate with application clients and do not require changes to application clients.

3. What is NAT?

Network Address Translation is a method by which IP addresses are mapped from one address realm to another, providing transparent routing to end hosts. There are many variations of address translation that lend themselves to different applications. However, all flavors of NAT devices should share the following characteristics.

RED HAT 1012
RED HAT V. CAS

a) Transparent Address assignment.
b) Transparent routing through address translation.
(routing here refers to forwarding packets, and not exchanging routing information)
c) ICMP error packet payload translation.

Below is a diagram illustrating a scenario in which NAT is enabled on a stub domain border router, connected to the Internet through a regional router made available by a service provider.

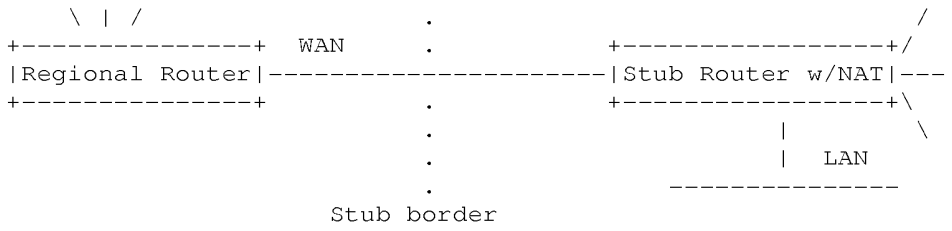


Figure 1: A typical NAT operation scenario

3.1. Transparent Address Assignment

NAT binds addresses in private network with addresses in global network and vice versa to provide transparent routing for the datagrams traversing between address realms. The binding in some cases may extend to transport level identifiers (such as TCP/UDP ports). Address binding is done at the start of a session. The following sub-sections describe two types of address assignments.

3.1.1. Static Address assignment

In the case of static address assignment, there is one-to-one address mapping for hosts between a private network address and an external network address for the lifetime of NAT operation. Static address assignment ensures that NAT does not have to administer address management with session flows.

3.1.2. Dynamic Address assignment

In this case, external addresses are assigned to private network hosts or vice versa, dynamically based on usage requirements and session flow determined heuristically by NAT. When the last session using an address binding is terminated, NAT would free the binding so that the global address could be recycled for later use. The exact nature of address assignment is specific to individual NAT implementations.

RED HAT 1012
RED HAT V. CAS

3.2. Transparent routing

A NAT router sits at the border between two address realms and translates addresses in IP headers so that when the packet leaves one realm and enters another, it can be routed properly. Because NAT devices have connections to multiple address realms, they must be careful to not improperly propagate information (e.g., via routing protocols) about networks from one address realm into another, where such an advertisement would be deemed unacceptable.

There are three phases to Address translation, as follows. Together these phases result in creation, maintenance and termination of state for sessions passing through NAT devices.

3.2.1. Address binding

Address binding is the phase in which a local node IP address is associated with an external address or vice versa, for purposes of translation. Address binding is fixed with static address assignments and is dynamic at session startup time with dynamic address assignments. Once the binding between two addresses is in place, all subsequent sessions originating from or to this host will use the same binding for session based packet translation.

New address bindings are made at the start of a new session, if such an address binding didn't already exist. Once a local address is bound to an external address, all subsequent sessions originating from the same local address or directed to the same local address will use the same binding.

The start of each new session will result in the creation of a state to facilitate translation of datagrams pertaining to the session. There can be many simultaneous sessions originating from the same host, based on a single address binding.

3.2.2. Address lookup and translation

Once a state is established for a session, all packets belonging to the session will be subject to address lookup (and transport identifier lookup, in some cases) and translation.

Address or transport identifier translation for a datagram will result in the datagram forwarding from the origin address realm to the destination address realm with network addresses appropriately updated.

RED HAT 1012
RED HAT V. CAS

3.2.3. Address unbinding

Address unbinding is the phase in which a private address is no longer associated with a global address for purposes of translation. NAT will perform address unbinding when it believes that the last session using an address binding has terminated. Refer section 2.6 for some heuristic ways to handle session terminations.

3.3. ICMP error packet translation

All ICMP error messages (with the exception of Redirect message type) will need to be modified, when passed through NAT. The ICMP error message types needing NAT modification would include Destination-Unreachable, Source-Quench, Time-Exceeded and Parameter-Problem. NAT should not attempt to modify a Redirect message type.

Changes to ICMP error message will include changes to the original IP packet (or portions thereof) embedded in the payload of the ICMP error message. In order for NAT to be completely transparent to end hosts, the IP address of the IP header embedded in the payload of the ICMP packet must be modified, the checksum field of the same IP header must correspondingly be modified, and the accompanying transport header. The ICMP header checksum must also be modified to reflect changes made to the IP and transport headers in the payload. Furthermore, the normal IP header must also be modified.

4.0. Various flavors of NAT

There are many variations of address translation that lend themselves to different applications. NAT flavors listed in the following sub-sections are by no means exhaustive, but they do capture the significant differences that abound.

The following diagram will be used as a base model to illustrate NAT flavors. Host-A, with address Addr-A is located in a private realm, represented by the network N-Pri. N-Pri is isolated from external network through a NAT router. Host-X, with address Addr-X is located in an external realm, represented by the network N-Ext. NAT router with two interfaces, each attached to one of the realms provides transparent routing between the two realms. The interface to the external realm is assigned an address of Addr-Nx and the interface to private realm is assigned an address of Addr-Np. Further, it may be understood that addresses Addr-A and Addr-Np correspond to N-Pri network and the addresses Addr-X and Addr-Nx correspond to N-Ext network.

RED HAT 1012
RED HAT V. CAS

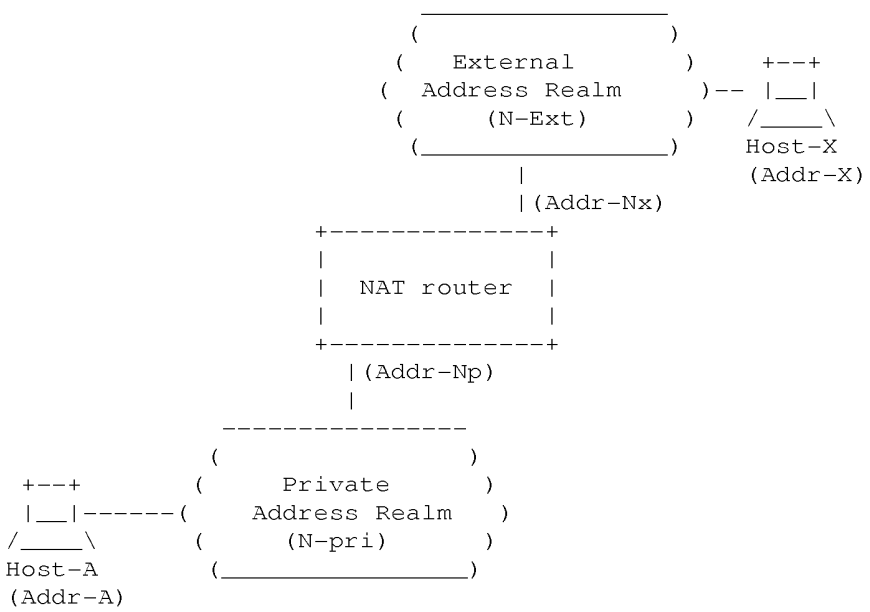


Figure 2: A base model to illustrate NAT terms.

4.1. Traditional NAT (or) Outbound NAT

Traditional NAT would allow hosts within a private network to transparently access hosts in the external network, in most cases. In a traditional NAT, sessions are uni-directional, outbound from the private network. This is in contrast with Bi-directional NAT, which permits sessions in both inbound and outbound directions. A detailed description of Bi-directional NAT may be found in section 4.2.

The following is a description of the properties of realms supported by traditional NAT. IP addresses of hosts in external network are unique and valid in external as well as private networks. However, the addresses of hosts in private network are unique only within the private network and may not be valid in the external network. In other words, NAT would not advertise private networks to the external realm. But, networks from the external realm may be advertised within the private network. The addresses used within private network must not overlap with the external addresses. Any given address must either be a private address or an external address; not both.

RED HAT 1012
RED HAT V. CAS

A traditional NAT router in figure 2 would allow Host-A to initiate sessions to Host-X, but not the other way around. Also, N-Ext is routable from within N-Pri, whereas N-Pri may not be routable from N-Ext.

Traditional NAT is primarily used by sites using private addresses that wish to allow outbound sessions from their site.

There are two variations to traditional NAT, namely Basic NAT and NAPT (Network Address Port Translation). These are discussed in the following sub-sections.

4.1.1. Basic NAT

With Basic NAT, a block of external addresses are set aside for translating addresses of hosts in a private domain as they originate sessions to the external domain. For packets outbound from the private network, the source IP address and related fields such as IP, TCP, UDP and ICMP header checksums are translated. For inbound packets, the destination IP address and the checksums as listed above are translated.

A Basic NAT router in figure 2 may be configured to translate N-Pri into a block of external addresses, say Addr-i through Addr-n, selected from the external network N-Ext.

4.1.2. Network Address Port Translation (NAPT)

NAPT extends the notion of translation one step further by also translating transport identifier (e.g., TCP and UDP port numbers, ICMP query identifiers). This allows the transport identifiers of a number of private hosts to be multiplexed into the transport identifiers of a single external address. NAPT allows a set of hosts to share a single external address. Note that NAPT can be combined with Basic NAT so that a pool of external addresses are used in conjunction with port translation.

For packets outbound from the private network, NAPT would translate the source IP address, source transport identifier and related fields such as IP, TCP, UDP and ICMP header checksums. Transport identifier can be one of TCP/UDP port or ICMP query ID. For inbound packets, the destination IP address, destination transport identifier and the IP and transport header checksums are translated.

RED HAT 1012
RED HAT V. CAS

A NAPT router in figure 2 may be configured to translate sessions originated from N-Pri into a single external address, say Addr-i.

Very often, the external interface address Addr-Nx of NAPT router is used as the address to map N-Pri to.

4.2. Bi-directional NAT (or) Two-Way NAT

With a Bi-directional NAT, sessions can be initiated from hosts in the public network as well as the private network. Private network addresses are bound to globally unique addresses, statically or dynamically as connections are established in either direction. The name space (i.e., their Fully Qualified Domain Names) between hosts in private and external networks is assumed to be end-to-end unique. Hosts in external realm access private realm hosts by using DNS for address resolution. A DNS-ALG must be employed in conjunction with Bi-Directional NAT to facilitate name to address mapping. Specifically, the DNS-ALG must be capable of translating private realm addresses in DNS Queries and responses into their external realm address bindings, and vice versa, as DNS packets traverse between private and external realms.

The address space requirements outlined for traditional NAT routers are applicable here as well.

A Bi-directional NAT router in figure 2 would allow Host-A to initiate sessions to Host-X, and Host-X to initiate sessions to Host-A. Just as with traditional NAT, N-Ext is routable from within N-Pri, but N-Pri may not be routable from N-Ext.

4.3. Twice NAT

Twice NAT is a variation of NAT in that both the source and destination addresses are modified by NAT as a datagram crosses address realms. This is in contrast to Traditional-NAT and Bi-Directional NAT, where only one of the addresses (either source or destination) is translated. Note, there is no such term as 'Once-NAT'.

Twice NAT is necessary when private and external realms have address collisions. The most common case where this would happen is when a site had (improperly) numbered its internal nodes using public addresses that have been assigned to another organization. Alternatively, a site may have changed from one provider to another, but chosen to keep (internally) the addresses it had been assigned by the first provider. That provider might then later reassign those addresses to someone else. The key issue in such cases is that the address of the host in the external realm may have been assigned the

RED HAT 1012
RED HAT V. CAS

same address as a host within the local site. If that address were to appear in a packet, it would be forwarded to the internal node rather than through the NAT device to the external realm. Twice-NAT attempts to bridge these realms by translating both source and destination address of an IP packet, as the packet transitions realms.

Twice-NAT works as follows. When Host-A wishes to initiate a session to Host-X, it issues a DNS query for Host-X. A DNS-ALG intercepts the DNS query, and in the response returned to Host-A the DNS-ALG replaces the address for Host-X with one that is properly routable in the local site (say Host-XPRIME). Host A then initiates communication with Host-XPRIME. When the packets traverse the NAT device, the source IP address is translated (as in the case of traditional NAT) and the destination address is translated to Host-X. A similar translation is performed on return packets coming from Host-X.

The following is a description of the properties of realms supported by Twice-NAT. Network address of hosts in external network are unique in external networks, but not within private network. Likewise, the network address of hosts in private network are unique only within the private network. In other words, the address space used in private network to locate hosts in private and public networks is unrelated to the address space used in public network to locate hosts in private and public networks. Twice NAT would not be allowed to advertise local networks to the external network or vice versa.

A Twice NAT router in figure 2 would allow Host-A to initiate sessions to Host-X, and Host-X to initiate sessions to Host-A. However, N-Ext (or a subset of N-Ext) is not routable from within N-Pri, and N-Pri is not routable from N-Ext.

Twice NAT is typically used when address space used in a Private network overlaps with addresses used in the Public space. For example, say a private site uses the 200.200.200.0/24 address space which is officially assigned to another site in the public internet. Host_A (200.200.200.1) in Private space seeks to connect to Host_X (200.200.200.100) in Public space. In order to make this connection work, Host_X's address is mapped to a different address for Host_A and vice versa. The twice NAT located at the Private site border may be configured as follows:

RED HAT 1012
RED HAT V. CAS

```
Private to Public : 200.200.200.0/24 -> 138.76.28.0/24
Public to Private : 200.200.200.0/24 -> 172.16.1.0/24

Datagram flow  : Host_A(Private) ->  Host_X(Public)

a) Within private network

   DA: 172.16.1.100      SA: 200.200.200.1

b) After twice-NAT translation

  DA: 200.200.200.100    SA: 138.76.28.1

Datagram flow Host_X (Public) -> Host_A (Private)

a) Within Public network

   DA: 138.76.28.1       SA: 200.200.200.100

b) After twice-NAT translation, in private network

   SA: 200.200.200.1     DA: 172.16.1.100
```

4.4. Multihomed NAT

There are limitations to using NAT. For example, requests and responses pertaining to a session must be routed via the same NAT router, as a NAT router maintains state information for sessions established through it. For this reason, it is often suggested that NAT routers be operated on a border router unique to a stub domain, where all IP packets are either originated from the domain or destined to the domain. However, such a configuration would turn a NAT router into a single point of failure.

In order for a private network to ensure that connectivity with external networks is retained even as one of the NAT links fail, it is often desirable to multihome the private network to same or multiple service providers with multiple connections from the private domain, be it from same or different NAT boxes.

For example, a private network could have links to two different providers and the sessions from private hosts could flow through the NAT router with the best metric for a destination. When one of NAT routers fail, the other could route traffic for all connections.

RED HAT 1012
RED HAT V. CAS

Multiple NAT boxes or multiple links on the same NAT box, sharing the same NAT configuration can provide fail-safe backup for each other. In such a case, it is necessary for backup NAT device to exchange state information so that a backup NAT can take on session load transparently when the primary NAT fails. NAT backup becomes simpler, when configuration is based on static maps.

5.0. Realm Specific IP (RSIP)

"Realm Specific IP" (RSIP) is used to characterize the functionality of a realm-aware host in a private realm, which assumes realm-specific IP address to communicate with hosts in private or external realm.

A "Realm Specific IP Client" (RSIP client) is a host in a private network that adopts an address in an external realm when connecting to hosts in that realm to pursue end-to-end communication. Packets generated by hosts on either end in such a setup would be based on addresses that are end-to-end unique in the external realm and do not require translation by an intermediary process.

A "Realm Specific IP Server" (RSIP server) is a node resident on both private and external realms, that can facilitate routing of external realm packets within a private realm. These packets may either have been originated by an RSIP client or directed to an RSIP-client. RSIP-Server may also be the same node that assigns external realm addresses to RSIP-Clients.

There are two variations to RSIP, namely Realm-specific Address IP (RSA-IP) and Realm-Specific Address and Port IP (RSAP-IP). These variations are discussed in the following sub-sections.

5.1. Realm Specific Address IP (RSA-IP)

A Realm Specific Address IP (RSA-IP) client adopts an IP address from the external address space when connecting to a host in external realm. Once an RSA-IP client assumes an external address, no other host in private or external domain can assume the same address, until that address is released by the RSA-IP client.

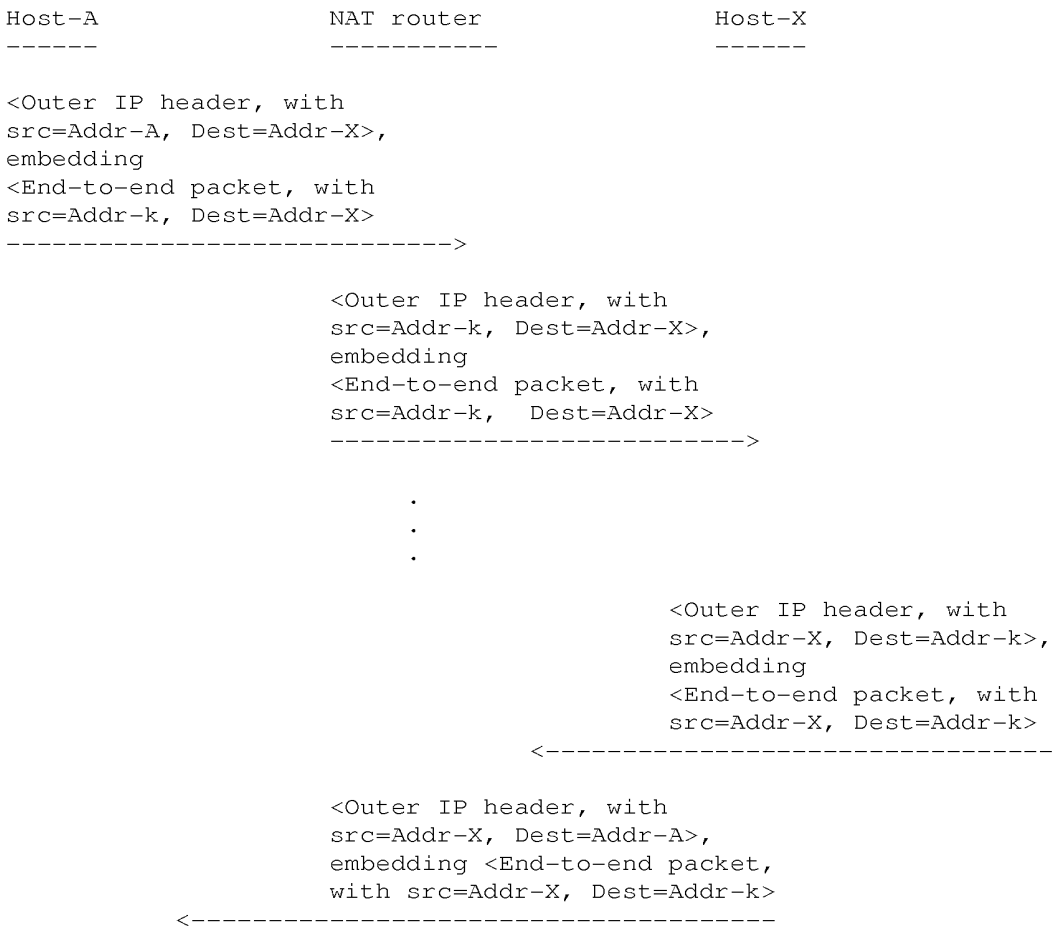
The following is a discussion of routing alternatives that may be pursued for the end-to-end RSA-IP packets within private realm. One approach would be to tunnel the packet to the destination. The outer header can be translated by NAT as normal without affecting the addresses used in the internal header. Another approach would be to set up a bi-directional tunnel between the RSA-IP Client and the border router straddling the two address realms. Packets to and from the client would be tunneled, but packets would be forwarded as

RED HAT 1012
RED HAT V. CAS

normal between the border router and the remote destination. Note, the tunnel from the client TO the border router may not be necessary. You might be able to just forward the packet directly. This should work so long as your internal network isn't filtering packets based on source addresses (which in this case would be external addresses).

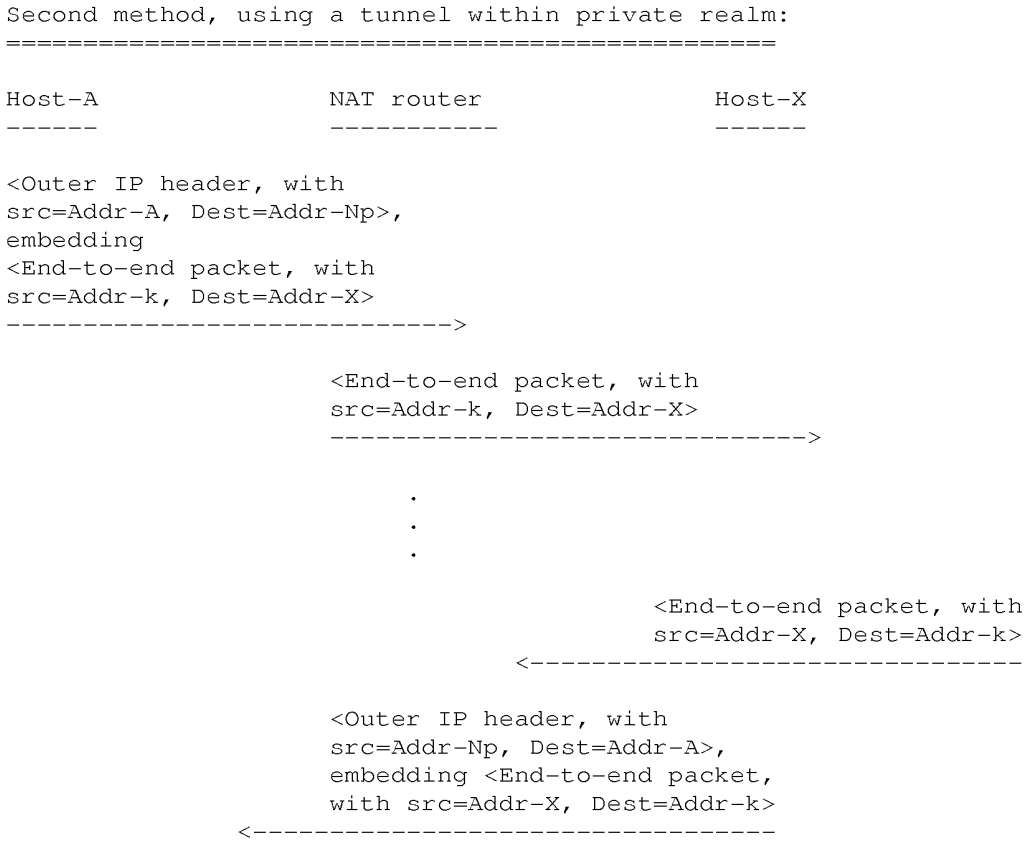
As an example, Host-A in figure 2 above, could assume an address Addr-k from the external realm and act as RSA-IP-Client to allow end-to-end sessions between Addr-k and Addr-X. Traversal of end-to-end packets within private realm may be illustrated as follows:

First method, using NAT router enroute to translate:



RED HAT 1012
RED HAT V. CAS

Second method, using a tunnel within private realm:



There may be other approaches to pursue.

An RSA-IP-Client has the following characteristics. The collective set of operations performed by an RSA-IP-Client may be termed "RSA-IP".

1. Aware of the realm to which its peer nodes belong.

2. Assumes an address from external realm when communicating with hosts in that realm. Such an address may be assigned statically or obtained dynamically (through a yet-to-be-defined protocol) from a node capable of assigning addresses from external realm. RSA-IP-Server could be the node coordinating external realm address assignment.

RED HAT 1012
RED HAT V. CAS

3. Route packets to external hosts using an approach amenable to RSA-IP-Server. In all cases, RSA-IP-Client will likely need to act as a tunnel end-point, capable of encapsulating end-to-end packets while forwarding and decapsulating in the return path.

"Realm Specific Address IP Server" (RSA-IP server) is a node resident on both private and external realms, that facilitates routing of external realm packets specific to RSA-IP clients inside a private realm. An RSA-IP-Server may be described as having the following characteristics.

1. May be configured to assign addresses from external realm to RSA-IP-Clients, either statically or dynamically.

2. Must be a router resident on both the private and external address realms.

3. Must be able to provide a mechanism to route external realm packets within private realm. Of the two approaches described, the first approach requires RSA-IP-Server to be a NAT router providing transparent routing for the outer header. This approach requires the external peer to be a tunnel end-point.

   With the second approach, an RSA-IP-Server could be any router (including a NAT router) that can be a tunnel end-point with RSA-IP-Clients. It would detunnel end-to-end packets outbound from RSA-IP-Clients and forward to external hosts. On the return path, it would locate RSA-IP-Client tunnel, based on the destination address of the end-to-end packet and encapsulate the packet in a tunnel to forward to RSA-IP-Client.

RSA-IP-Clients may pursue any of the IPsec techniques, namely transport or tunnel mode Authentication and confidentiality using AH and ESP headers on the embedded packets. Any of the tunneling techniques may be adapted for encapsulation between RSA-IP-Client and RSA-IP-Server or between RSA-IP-Client and external host. For example, IPsec tunnel mode encapsulation is a valid type of encapsulation that ensures IPsec authentication and confidentiality for the embedded end-to-end packets.

5.2 Realm Specific Address and port IP (RSAP-IP)

Realm Specific Address and port IP (RSAP-IP) is a variation of RSIP in that multiple private hosts use a single external address, multiplexing on transport IDentifiers (i.e., TCP/UDP port numbers and ICMP Query IDs).

RED HAT 1012
RED HAT V. CAS

"RSAP-IP-Client" may be defined similar to RSA-IP-Client with the variation that RSAP-IP-Client assumes a tuple of (external address, transport Identifier) when connecting to hosts in external realm to pursue end-to-end communication. As such, communication with external nodes for an RSAP-IP-Client may be limited to TCP, UDP and ICMP sessions.

"RSAP-IP-Server" is similar to RSA-IP-Server in that it facilitates routing of external realm packets specific to RSAP-IP clients inside a private realm. Typically, an RSAP-IP-Server would also be the one to assign transport tuples to RSAP-IP-Clients.

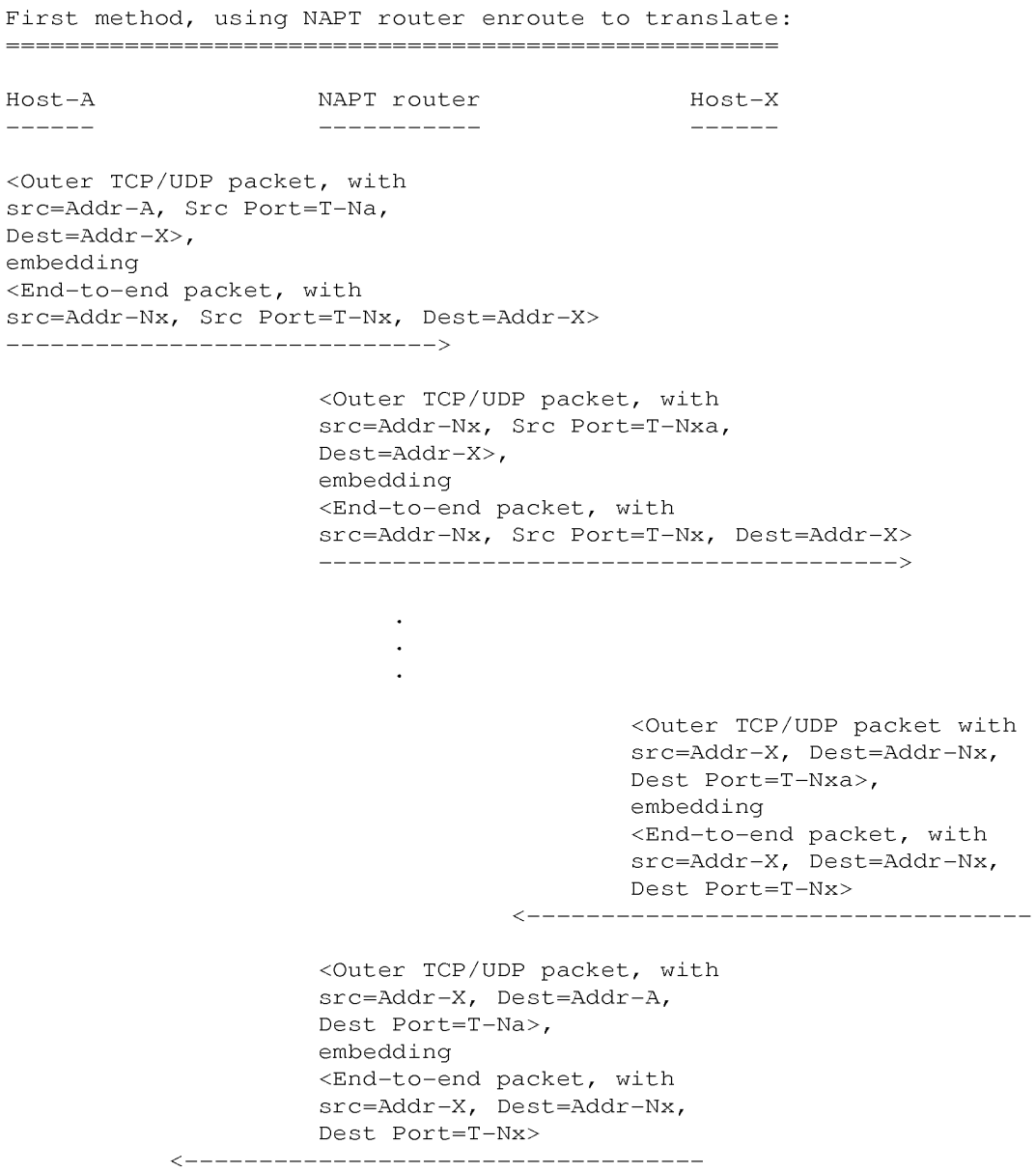
A NAPT router enroute could serve as RSAP-IP-Server, when the outer encapsulation is TCP/UDP based and is addressed between the RSAP-IP-Client and external peer. This approach requires the external peer to be the end-point of TCP/UDP based tunnel. Using this approach, RSAP-IP-Clients may pursue any of the IPsec techniques, namely transport or tunnel mode authentication and confidentiality using AH and ESP headers on the embedded packets. Note however, IPsec tunnel mode is not a valid type of encapsulation, as a NAPT router cannot provide routing transparency to AH and ESP protocols.

Alternately, packets may be tunneled between RSAP-IP-Client and RSAP-IP-Server such that RSAP-IP-Server would detunnel packets outbound from RSAP-IP-Clients and forward to external hosts. On the return path, RSAP-IP-Server would locate RSAP-IP-Client tunnel, based on the tuple of (destination address, transport Identifier) and encapsulate the original packet within a tunnel to forward to RSAP-IP-Client. With this approach, there is no limitation on the tunneling technique employed between RSAP-IP-Client and RSAP-IP-Server. However, there are limitations to applying IPsec based security on end-to-end packets. Transport mode based authentication and integrity may be attained. But, confidentiality cannot be permitted because RSAP-IP-Server must be able to examine the destination transport Identifier in order to identify the RSAP-IP-tunnel to forward inbound packets to. For this reason, only the transport mode TCP, UDP and ICMP packets protected by AH and ESP-authentication can traverse a RSAP-IP-Server using this approach.

As an example, say Host-A in figure 2 above, obtains a tuple of (Addr-Nx, TCP port T-Nx) from NAPT router to act as RSAP-IP-Client to initiate end-to-end TCP sessions with Host-X. Traversal of end-to-end packets within private realm may be illustrated as follows. In the first method, outer layer of the outgoing packet from Host-A uses (private address Addr-A, source port T-Na) as source tuple to communicate with Host-X. NAPT router enroute translates this tuple into (Addr-Nx, Port T-Nxa). This translation is independent of RSAP-IP-Client tuple parameters used in the embedded packet.

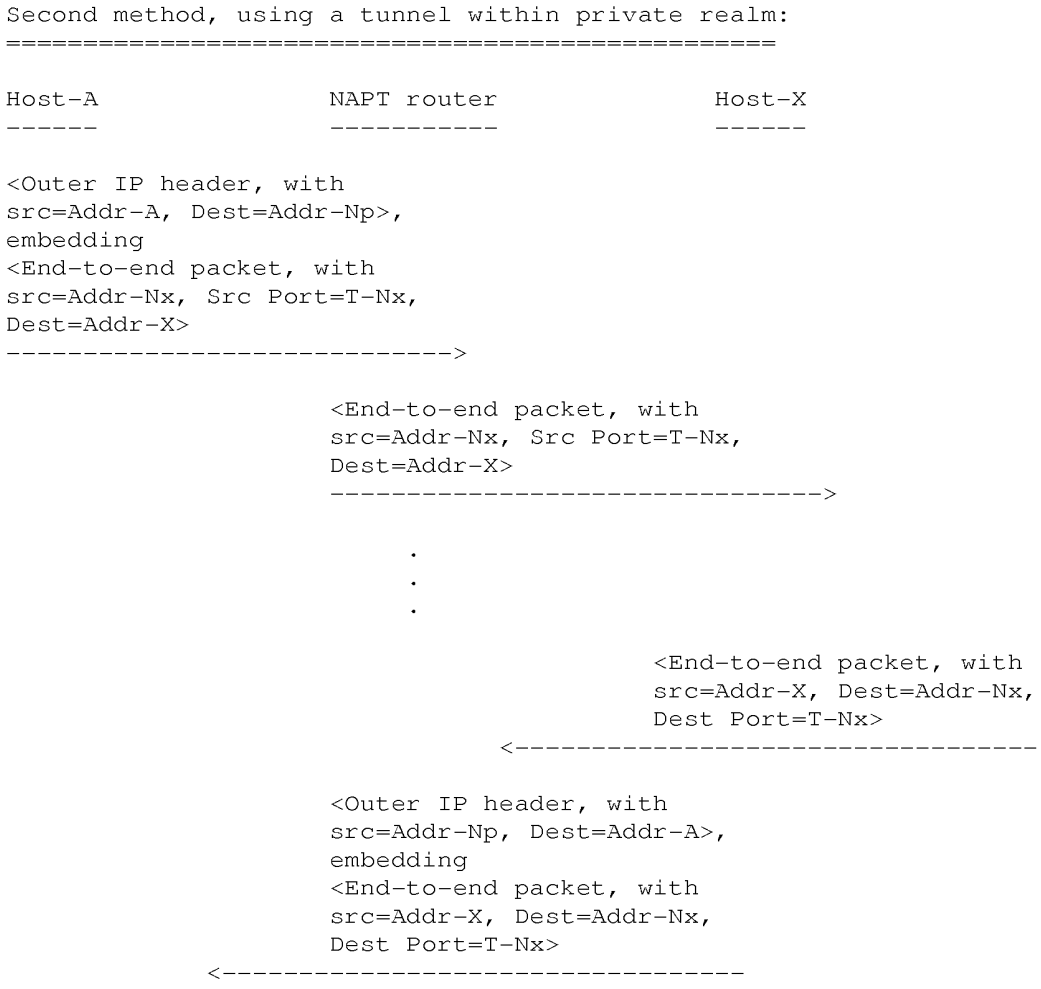
RED HAT 1012
RED HAT V. CAS

First method, using NAPT router enroute to translate:



RED HAT 1012
RED HAT V. CAS

Second method, using a tunnel within private realm:



6.0. Private Networks and Tunnels

Let us consider the case where your private network is connected to the external world via tunnels. In such a case, tunnel encapsulated traffic may or may not contain translated packets depending upon the characteristics of address realms a tunnel is bridging.

The following subsections discuss two scenarios where tunnels are used (a) in conjunction with Address translation, and (b) without translation.

RED HAT 1012
602
RED HAT V. CAS

6.1. Tunneling translated packets

All variations of address translations discussed in the previous section can be applicable to direct connected links as well as tunnels and virtual private networks (VPNs).

For example, a private network connected to a business partner through a VPN could employ traditional NAT to communicate with the partner. Likewise, it is possible to employ twice NAT, if the partner's address space overlapped with the private network. There could be a NAT device on one end of the tunnel or on both ends of the tunnel. In all cases, traffic across the VPN can be encrypted for security purposes. Security here refers to security for traffic across VPNs alone. End-to-end security requires trusting NAT devices within private network.

6.2. Backbone partitioned private Networks

There are many instances where a private network (such as a corporate network) is spread over different locations and use public backbone for communications between those locations. In such cases, it is not desirable to do address translation, both because large numbers of hosts may want to communicate across the backbone, thus requiring large address tables, and because there will be more applications that depend on configured addresses, as opposed to going to a name server. We call such a private network a backbone-partitioned private network.

Backbone-partitioned stubs should behave as though they were a non-partitioned stub. That is, the routers in all partitions should maintain routes to the local address spaces of all partitions. Of course, the (public) backbones do not maintain routes to any local addresses. Therefore, the border routers must tunnel (using VPNs) through the backbones using encapsulation. To do this, each NAT box will set aside a global address for tunneling.

When a NAT box x in stub partition X wishes to deliver a packet to stub partition Y, it will encapsulate the packet in an IP header with destination address set to the global address of NAT box y that has been reserved for encapsulation. When NAT box y receives a packet with that destination address, it decapsulates the IP header and routes the packet internally. Note, there is no address translation in the process; merely transfer of private network packets over an external network tunnel backbone.

RED HAT 1012
603
RED HAT V. CAS

7.0. NAT operational characteristics

NAT devices are application unaware in that the translations are limited to IP/TCP/UDP/ICMP headers and ICMP error messages only. NAT devices do not change the payload of the packets, as payloads tend to be application specific.

NAT devices (without the inclusion of ALGs) do not examine or modify transport payload. For this reason, NAT devices are transparent to applications in many cases. There are two areas, however, where NAT devices often cause difficulties: 1) when an application payload includes an IP address, and 2) when end-to-end security is needed. Note, this is not a comprehensive list.

Application layer security techniques that do not make use of or depend on IP addresses will work correctly in the presence of NAT (e.g., TLS, SSL and ssh). In contrast, transport layer techniques such as IPSec transport mode or the TCP MD5 Signature Option RFC 2385 [Ref 17] do not.

In IPSec transport mode, both AH and ESP have an integrity check covering the entire payload. When the payload is TCP or UDP, the TCP/UDP checksum is covered by the integrity check. When a NAT device modifies an address the checksum is no longer valid with respect to the new address. Normally, NAT also updates the checksum, but this is ineffective when AH and ESP are used. Consequently, receivers will discard a packet either because it fails the IPSec integrity check (if the NAT device updates the checksum), or because the checksum is invalid (if the NAT device leaves the checksum unmodified).

Note that IPsec tunnel mode ESP is permissible so long as the embedded packet contents are unaffected by the outer IP header translation. Although this technique does not work in traditional NAT deployments (i.e., where hosts are unaware that NATs are present), the technique is applicable to Realm-Specific IP as described in Section 5.0.

Note also that end-to-end ESP based transport mode authentication and confidentiality are permissible for packets such as ICMP, whose IP payload content is unaffected by the outer IP header translation.

NAT devices also break fundamental assumptions by public key distribution infrastructures such as Secure DNS RFC 2535 [Ref 18] and X.509 certificates with signed public keys. In the case of Secure

RED HAT 1012
RED HAT V. CAS

DNS, each DNS RRset is signed with a key from within the zone. Moreover, the authenticity of a specific key is verified by following a chain of trust that goes all the way to the DNS root. When a DNS-ALG modifies addresses (e.g., as in the case of Twice-NAT), verification of signatures fails.

It may be of interest to note that IKE (Session key negotiation protocol) is a UDP based session layer protocol and is not protected by network based IPsec security. Only a portion of the individual payloads within IKE are protected. As a result, IKE sessions are permissible across NAT, so long as IKE payload does not contain addresses and/or transport IDs specific to one realm and not the other. Given that IKE is used to setup IPSec associations, and there are at present no known ways of making IPSec work through a NAT function, it is a future work item to take advantage of IKE through a NAT box.

One of the most popular internet applications "FTP" would not work with the definition of NAT as described. The following sub-section is devoted to describing how FTP is supported on NAT devices. FTP ALG is an integral part of most NAT implementations. Some vendors may choose to include additional ALGs to custom support other applications on the NAT device.

7.1. FTP support

"PORT" command and "PASV" response in FTP control session payload identify the IP address and TCP port that must be used for the data session it supports. The arguments to the PORT command and PASV response are an IP address and a TCP port in ASCII. An FTP ALG is required to monitor and update the FTP control session payload so that information contained in the payload is relevant to end nodes. The ALG must also update NAT with appropriate data session tuples and session orientation so that NAT could set up state information for the FTP data sessions.

Because the address and TCP port are encoded in ASCII, this may result in a change in the size of packet. For instance, 10,18,177,42,64,87 is 18 ASCII characters, whereas 193,45,228,137,64,87 is 20 ASCII characters. If the new size is same as the previous, only the TCP checksum needs adjustment as a result of change of data. If the new size is less than or greater than the previous, TCP sequence numbers must also be changed to reflect the change in length of FTP control data portion. A special table may be used by the ALG to correct the TCP sequence and acknowledge numbers. The sequence number and acknowledgement correction will need to be performed on all future packet of the connection.

RED HAT 1012
605
RED HAT V. CAS

8.0. NAT limitations

8.1. Applications with IP-address Content

Not All applications lend themselves easily to address translation by NAT devices. Especially, the applications that carry IP address (and TU port, in case of NAPT) inside the payload. Application Level Gateways, or ALGs must be used to perform translations on packets pertaining to such applications. ALGs may optionally utilize address (and TU port) assignments made by NAT and perform translations specific to the application. The combination of NAT functionality and ALGs will not provide end-to-end security assured by IPsec. However, tunnel mode IPsec can be accomplished with NAT router serving as tunnel end point.

SNMP is one such application with address content in payload. NAT routers would not translate IP addresses within SNMP payloads. It is not uncommon for an SNMP specific ALG to reside on a NAT router to perform SNMP MIB translations proprietary to the private network.

8.2. Applications with inter-dependent control and data sessions

NAT devices operate on the assumption that each session is independent. Session characteristics like session orientation, source and destination IP addresses, session protocol, and source and destination transport level identifiers are determined independently at the start of each new session.

However, there are applications such as H.323 that use one or more control sessions to set the characteristics of the follow-on sessions in their control session payload. Such applications require use of application specific ALGs that can interpret and translate the payload, if necessary. Payload interpretation would help NAT be prepared for the follow-on data sessions.

8.3. Debugging Considerations

NAT increases the probability of mis-addressing. For example, same local address may be bound to different global address at different times and vice versa. As a result, any traffic flow study based purely on global addresses and TU ports could be confused and might misinterpret the results.

If a host is abusing the Internet in some way (such as trying to attack another machine or even sending large amounts of junk mail or something) it is more difficult to pinpoint the source of the trouble because the IP address of the host is hidden in a NAT router.

RED HAT 1012
RED HAT V. CAS

8.4. Translation of fragmented FTP control packets

Translation of fragmented FTP control packets is tricky when the packets contain "PORT" command or response to "PASV" command. Clearly, this is a pathological case. NAT router would need to assemble the fragments together first and then translate prior to forwarding.

Yet another case would be when each character of packets containing "PORT" command or response to "PASV" is sent in a separate datagram, unfragmented. In this case, NAT would simply have to let the packets through, without translating the TCP payload. Of course, the application will fail if the payload needed to be altered. The application could still work in a few cases, where the payload contents can be valid in both realms, without modifications enroute. For example, FTP originated from a private host would still work while traversing a traditional NAT or bi-directional NAT device, so long as the FTP control session employed PASV command to establish data sessions. The reason being that the address and port number specified by FTP server in the PASV response (sent as multiple unfragmented packets) is valid to the private host, as is. The NAT device will simply view the ensuing data session (also originating from private host) as an independent TCP session.

8.5. Compute intensive

NAT is compute intensive even with the help of a clever checksum adjustment algorithm, as each data packet is subject to NAT lookup and modifications. As a result, router forwarding throughput could be slowed considerably. However, so long as the processing capacity of the NAT device exceeds line processing rate, this should not be a problem.

9.0. Security Considerations

Many people view traditional NAT router as a one-way (session) traffic filter, restricting sessions from external hosts into their machines. In addition, when address assignment in NAT router is done dynamically, that makes it harder for an attacker to point to any specific host in the NAT domain. NAT routers may be used in conjunction with firewalls to filter unwanted traffic.

If NAT devices and ALGs are not in a trusted boundary, that is a major security problem, as ALGs could snoop end user traffic payload. Session level payload could be encrypted end to end, so long as the payload does not contain IP addresses and/or transport identifiers that are valid in only one of the realms. With the exception of RSIP, end-to-end IP network level security assured by current IPsec

RED HAT 1012
607
RED HAT V. CAS

techniques is not attainable with NAT devices in between. One of the ends must be a NAT box. Refer section 7.0 for a discussion on why end-to-end IPsec security cannot be assured with NAT devices along the route.

The combination of NAT functionality, ALGs and firewalls will provide a transparent working environment for a private networking domain. With the exception of RSIP, end-to-end network security assured by IPsec cannot be attained for end-hosts within the private network (Refer section 5.0 for RSIP operation). In all other cases, if you want to use end-to-end IPsec, there cannot be a NAT device in the path. If we make the assumption that NAT devices are part of a trusted boundary, tunnel mode IPsec can be accomplished with NAT router (or a combination of NAT, ALGs and firewall) serving as tunnel end point.

NAT devices, when combined with ALGs, can ensure that the datagrams injected into Internet have no private addresses in headers or payload. Applications that do not meet these requirements may be dropped using firewall filters. For this reason, it is not uncommon to find NAT, ALG and firewall functions co-exist to provide security at the borders of a private network. NAT gateways can be used as tunnel end points to provide secure VPN transport of packet data across an external network domain.

Below are some additional security considerations associated with NAT routers.

1. UDP sessions are inherently unsafe. Responses to a datagram could come from an address different from the target address used by sender ([Ref 4]). As a result, an incoming UDP packet might match the outbound session of a traditional NAT router only in part (the destination address and UDP port number of the packet match, but the source address and port number may not). In such a case, there is a potential security compromise for the NAT device in permitting inbound packets with partial match. This UDP security issue is also inherent to firewalls.

   Traditional NAT implementations that do not track datagrams on a per-session basis but lump states of multiple UDP sessions using the same address binding into a single unified session could compromise the security even further. This is because, the granularity of packet matching would be further limited to just the destination address of the inbound UDP packets.

2. Multicast sessions (UDP based) are another source for security weakness for traditional-NAT routers. Once again, firewalls face the same security dilemma as the NAT routers.

RED HAT 1012
608
RED HAT V. CAS

Say, a host on private network initiated a multicast session. Datagram sent by the private host could trigger responses in the reverse direction from multiple external hosts. Traditional-NAT implementations that use a single state to track a multicast session cannot determine for certain if the incoming UDP packet is in response to an existing multicast session or the start of new UDP session initiated by an attacker.

3. NAT devices can be a target for attacks.

   Since NAT devices are Internet hosts they can be the target of a number of different attacks, such as SYN flood and ping flood attacks. NAT devices should employ the same sort of protection techniques as Internet-based servers do.

REFERENCES

[1] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot,G. and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[2] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October, 1994.

[3] Braden, R., "Requirements for Internet Hosts -- Communication Layers", STD 3, RFC 1122, October 1989.

[4] Braden, R., "Requirements for Internet Hosts -- Application and Support", STD 3, RFC 1123, October 1989.

[5] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

[6] Postel, J. and J. Reynolds, "File Transfer Protocol (FTP)", STD 9, RFC 959, October 1985.

[7] Postel, J., "Transmission Control Protocol (TCP) Specification", STD 7, RFC 793, September 1981.

[8] Postel, J., "Internet Control Message Protocol Specification" STD 5, RFC 792, September 1981.

[9] Postel, J., "User Datagram Protocol (UDP)", STD 6, RFC 768, August 1980.

[10] Mogul, J. and J. Postel, "Internet Standard Subnetting Procedure", STD 5, RFC 950, August 1985.

RED HAT 1012
RED HAT V. CAS

[11] Carpenter, B., Crowcroft, J. and Y. Rekhter, "IPv4 Address Behavior Today", RFC 2101, February 1997.

[12] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[13] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[14] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[15] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[16] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[17] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[18] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

Authors' Addresses

Pyda Srisuresh
Lucent Technologies
4464 Willow Road
Pleasanton, CA 94588-8519
U.S.A.

Phone: (925) 737-2153
Fax: (925) 737-2110
EMail: srisuresh@lucent.com

Matt Holdrege
Lucent Technologies
1701 Harbor Bay Parkway
Alameda, CA 94502

Phone: (510) 769-6001
EMail: holdrege@lucent.com

RED HAT 1012
RED HAT V. CAS

Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

RED HAT 1012
611
RED HAT V. CAS

A Simulation Study of Packet Path Diversity for TCP File Transfer and Media Transport on the Internet

Eckehard G. Steinbach *[1], Yi J. Liang [2], and Bernd Girod [2]

[1] Institute of Communication Networks
Munich University of Technology
Steinbach@ei.tum.de

[2] Information Systems Laboratory
Department of Electrical Engineering
Stanford University

ABSTRACT

We study the performance of TCP-based media transport when combined with packet path diversity (PPD) in the Internet. We present a simple PPD-TCP transport scheme where the acknowledgments (ACKs) from the receiver are duplicated and sent across independent network paths. We also investigate a PPD-TCP scheme where in addition to path-diversified ACKs on the backward path, packet path diversity is also used on the forward path. In order to keep the additional network load small, only small probe packets that carry the same TCP header information but not the payload are sent across the alternative path. From the time difference between the arrival of the probe packet and the corresponding data packet at the receiver we derive the superior path and, if advantageous, switch the path for the data packets. For comparison purposes we also implement a PPD-TCP transmission scheme where both, acknowledgment and data packets, are duplicated and sent over different network paths. This scheme provides us with an upper bound of the performance improvements that we can obtain compared to a corresponding single-path TCP transmission. The different PPD-TCP schemes are compared to standard single-path TCP transmission using ***ns*** *network simulations. It is shown that all investigated PPD-TCP schemes greatly outperform standard single-path TCP transmission for time-critical file transfer and media delivery.*

1 INTRODUCTION

Media content delivery such as video streaming over the Internet either uses a guaranteed-delivery transport protocol such as TCP or a partially reliable protocol based on UDP and RTP combined with proprietary ARQ, flow control, and congestion control mechanisms [12],[13]. While UDP-based transport makes the transmission process more continuous, one advantage of TCP-based transport is that the receiver is guaranteed to successfully receive all media packets which is beneficial if the media file is either to be stored after playout or to be replayed several times locally. Modern video encoding schemes like MPEG-2, MPEG-4, or H.263 introduce strong dependencies between successive packets. Hence, continuous video playout can only be guaranteed if all packets are available for decoding. Another advantage of TCP-based media delivery systems is the ease of implementation since no additional flow control, congestion control, or ARQ mechanisms have to be implemented. A simple HTTP-based web server can therefore be used to deliver the encoded media content as for instance demonstrated by the early streaming media program VivoActive 1.0 [16]. Most importantly, however, TCP-based media delivery naturally leads to transmission friendliness with respect to other flows that are competing for the same network resources.

One reason why TCP is not often used for real-time or near real-time media delivery is the potentially substantial fluctuation in the delivery times for the fragments of the encoded data, which is partly due to the lack of control over the rate at which the server pushes data to the client. Another reason is that TCP congestion control actively reacts to network load variations with the intention to share resources in a fair manner with other flows. This can also lead to significantly varying instantaneous throughput.

In this work, we investigate the combination of the TCP transport protocol with path-diversified packet transmission in the Internet. We show that this approach overcomes some of the aforementioned limitations of traditional single-path TCP-based media transport. For transmission with packet path diversity we assume that packets can be sent along more than one network path between server and client. Without loss of generality, we assume in the following path diversity with two network paths. The diversity gain that can be observed depends on the correlation of congestion, delay, and loss events on the two paths. For uncorrelated paths, excessively delayed or dropped packets should occur independently.

In order to illustrate the benefit of path-diversified TCP-based media transport, let us consider, for instance, temporary congestion on the path from the client to the

*This work has been supported, in part, by a gift from Hewlett Packard, and, in part, by the Stanford Networking Research Center while the first author was at Stanford University.

RED HAT 1012
RED HAT V. CAS

server which may lead to dropped acknowledgment packets. If an acknowledgment is not received, TCP automatically retransmits the data segment after the retransmission timer expires and increases the time until the next retransmission attempt is scheduled. This reaction unnecessarily reduces throughput if the forward path is not affected by the congestion. Sending the acknowledgment information along two independent network paths increases the probability that the acknowledgment information is received by the server and therefore reduces undesired throughput variations.

This paper is organized as follows. The next section briefly discusses prior work on multi-path media transport. In Section 3 we describe three different implementations of TCP transport with packet path diversity. In Section 4 we discuss the establishment of independent network paths in the Internet. Next, in Section 5 we present *ns* simulations where we compare the performance of the proposed PPD-TCP variants to traditional single-path TCP transport for various network topologies and network loads.

2 Related work

Almost 30 years ago, transmission with path diversity was proposed for load balancing, the reduction of transmission times, and fault tolerance purposes in store and forward networks by Maxemchuck [11]. More recent examples of multi-path transmission include [5],[9],[15]. A literature survey on multi-path transmission can be found in [8].

For media transport, multi-path transmission of complementary descriptions of a video signal has been considered in [2],[7]. In [3] the concept of transmission with path-diversity is considered in the context of media content delivery networks where a client can stream simultaneously from more than one content server. Significantly reduced infrastructure requirements for multi-path content delivery in comparison to single-path delivery are reported in [3]. In [10] packet path diversity augmented UDP-transport has been used for real-time voice transmission over the Internet and significant reductions in end-to-end latency and effective packet loss are observed.

The combination of TCP transport with path diversity for media delivery has received relatively little attention. One reason for this might be that most real-time or near real-time media transport in today's Internet is UDP-based. Nevertheless, the transmission of media content using TCP as a transport protocol has various advantages as discussed in the previous section which makes it worthwhile to study this combination. Prior work on path-diversified TCP-transport includes [6] where non real-time data transport over multiple paths in the context of ad-hoc networks has been proposed. Two different approaches for path-diversified TCP-transport are discussed in [6]. In the first approach, the data file transfer is split into multiple independent TCP-connections at a so called Meta-TCP layer. Since individual TCP-windows are used for each TCP-connection, the receiver has to resequence the corresponding subflows in a resequencing buffer. The second approach assumes path-diversity on the IP-layer which means that individual IP packets can be routed along different paths and one TCP-window is used.

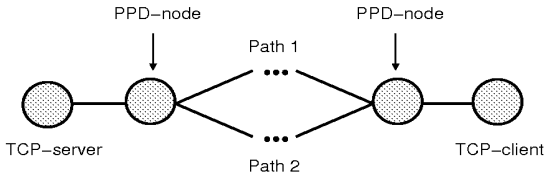


Figure 1: Packet path diversity with two PPD-nodes and two independent network paths.

3 TCP with Packet Path Diversity

In this section we first discuss various general issues involved when combining TCP transport with packet path diversity. We then describe the three PPD-TCP variants investigated in this work.

Fig. 1 shows the general architecture for two-path PPD-TCP. From one PPD-node to the other, two network paths exist. We assume in the following that both PPD-nodes can control the path along which a packet has to travel. The reason for separating the TCP hosts and the PPD hosts in Fig. 1 is to illustrate that the TCP server and client implementations remain unmodified in our study. The brute force way of exploiting packet path diversity would be to duplicate all packets at the PPD-nodes and to send one copy across each path. The corresponding receiving PPD-node would then simply use the earlier copy and discard the second copy. This is, of course, a significant waste of network resources but will serve as an upper performance bound for PPD-TCP in our study. If the PPD-node is not the final destination of the TCP connection, i.e., the TCP host and the PPD host differ, the earlier copy would be forwarded to the corresponding TCP host. This ensures that the transmission looks like a normal TCP connection from the perspective of the TCP server and client. In this scenario we can think of the PPD-node as a path diversity enabled router that can duplicate packets and send those packets along two different network paths. Obviously, this functionality could also be implemented on the TCP hosts themselves. The PPD implementation would then interface the TCP protocol. In this case at least the last link would typically be shared by both paths. The advantage of moving the PPD-nodes away from the edge of the network is that the two paths can be established without sharing last-mile links. If the number of shared links is minimized, we expect the congestion and loss characteristics of the two paths to be less correlated. We now discuss the three PPD-TCP variants implemented and compared in this work.

3.1 TCP with PPD ACKs

In this version of PPD-TCP only the acknowledgments (ACKs) from the client to the server are duplicated and sent across two different network paths. The data packets from the server to the client travel on the default path determined by the routing tables. If both copies of the ACK arrive at the PPD-node that is close to the server, the copy that arrives first is forwarded to the server while

RED HAT 1012
RED HAT V. CAS

the second copy is dropped. Discarding the second copy of the ACK is necessary to avoid the arrival of duplicate ACKs which would trigger the fast retransmission mode of TCP [14].

In single-path or standard TCP transport it can easily happen during congestion that acknowledgments are dropped by a router with a full buffer on their way to the server. The server has to assume that the corresponding data portion has not been successfully received by the client until a later ACK acknowledges this data fragment. With path diversity the probability that one of the two identical ACK packets will make it to the sender increases. The effect on the current throughput can be quite dramatic since TCP reacts to the lost ACKs by reducing its window size although the forward path might be of good quality. By path-diversifying the ACKs, TCP only has to handle congestion if it actually occurs on the forward path.

The overhead introduced by duplicating the acknowledgment packets is small. If we assume that for every 1500 Byte data packet two instead of one 40 Byte ACK packets are returned, we encounter an overhead of $(40)/(1500+40) = 2.6\%$.

3.2 PPD ACKs and PPD data packets

As mentioned at the beginning of Section 3, the maximum path diversity gain can be expected if copies of both, data and acknowledgment packets, are sent along two independent network paths. In Fig. 1 this would mean that the PPD-nodes duplicate the data packets from the TCP server to the client as well as the ACKs from the TCP client to the server. As before, the receiving PPD-nodes forward the copy that arrives earlier while discarding the later one. From the perspective of one single TCP flow this scheme will provide the maximum path diversity gain at the expense of penalizing all other flows competing for the same network resources. Obviously, this scheme results in sending twice as much data across the network (100% overhead). Another issue is that TCP congestion control will not react properly to congestion on the data path since transmission with one congested and one uncongested path will look like a single good path to TCP. As mentioned before we investigate this version of PPD-TCP only for the sake of comparison and use it as an upper performance bound. Please note that this form of TCP-PPD could be used to prioritize single TCP flows. This, however, is not the focus of this paper.

3.3 PPD ACKs and PPD probe packets

A more practical version of PPD-TCP that exploits path diversity in both directions generates a small probe packet for each data packet and sends data and probe packets along different network paths. In the following we refer to the path currently being used for data packets as the *active path*, while the other path used for probe packets is refered to as the *inactive path*. The probe packets are used to test the current transmission quality of the inactive path. Probe packets are small packets that carry just enough information for the PPD-node to uniquely identify a TCP data packet and its corresponding probe packet. Hence, the overhead introduced by packet path diversity remains small. At the receiving PPD-node we record the arrival time of both, data and corresponding probe packet, and use the time difference between the two to infer which path is currently better. In case the active data path is inferior to the inactive path we signal the other PPD-node to switch the paths and to send the data packets from now on along the inactive path which then becomes the active path. Assuming that the size of a probe or ACK packet is 40 Byte, the overhead we encounter for this version of PPD-TCP is $(40+40)/(1500+40) = 5.2\%$.

An interesting issue that arises in this context is how to decide when to switch the two paths. We have implemented a simple scheme that compares the quality of the two paths based on the arrival times of the probe and the corresponding data packet at the PPD node. In case a data packet arrives before the probe packet we assume the active path to be superior. If a probe packet arrives before we receive the corresponding data packet, we start a timer of duration ΔT. After this timer expires we check if meanwhile the corresponding data packet has been received. If yes, again the active path is assumed to be better. If no, we assume that the path that carries the probe packet is better and signal the PPD-node to switch paths. The probe packets are discarded as soon as they have served their purpose. The TCP data packets are always forwarded by the PPD-node to the TCP client.

ΔT is the additional delay that we would observe for a data packet if it were sent along the inactive path instead of a probe packet. This *virtual delay* for a data packet on the inactive path has to be inferred from the measured delay of the probe packet. The delay difference ΔT is mainly caused by the different transmission times of packets of different sizes.

In order to estimate ΔT, we send a packet pair along the alternative path at the start of a transmission session. The packet pair consists of a probe packet immediately followed by a data packet. The difference of the arrival times of the two packets is measured, which is the estimate of ΔT. During normal operation, only probe packets are sent over the inactive channel and the virtual arrival time of the data packets is computed by adding our most recently measured value of ΔT to the arrival time of the probe packet. This virtual delay can now be compared to the measured delay of the data packet that is sent along the active channel.

The switching between the paths so far is based on only one packet pair. This is obviously the most simple scheme. In future work we plan to regularly estimate ΔT during the media file transfer and to keep a short history of these measurements and base our decision to switch paths for instance on a majority vote of these recorded values. In this way, path changes caused by single packets that are extremely delayed or dropped and do not reflect the actual current load level on the paths can be avoided.

4 Internet Packet Path Diversity

In order to maximize the benefits of path diversity we have to select paths that exhibit largely uncorrelated

RED HAT 1012
RED HAT V. CAS

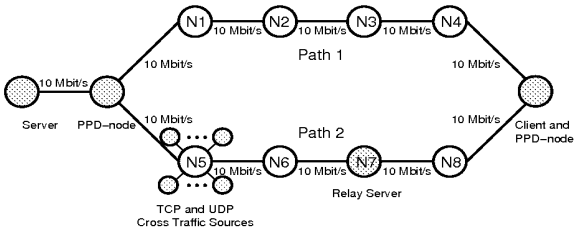


Figure 2: Multi-hop topology for *ns* network simulations. Each of the intermediate nodes N1 through N8 has a number of TCP data sources and UDP traffic sources attached (only shown for node N5).

transmission characteristics. With today's Internet protocols, the path a packet takes across the Internet is a function of its source and destination IP addresses as well as the entries of the routing tables involved. Selecting a specific path for a packet is largely unsupported in today's infrastructure.

As discussed in [2], IPv4 source routing is usually turned off within the Internet for security reasons. It would be more promising to implement path diversity by means of an overlay network that consists of relay nodes [2],[1]. Here, packets can be sent along different routes by encapsulating them into IP packets that have the addresses of different relay nodes as their destination. At the relay nodes, packets are forwarded to other relay nodes such that the packets travel along as few common links as possible.

With the next-generation IP protocol IPv6, the source node has a great amount of control over each packet's route. IPv6's loose source routing (LSR) allows packets to be sent via specified intermediate nodes. If widely implemented, this source routing feature of IPv6 will make future implementation of transmission with packet path diversity even simpler.

5 Simulation Results

In order to investigate the performance of TCP with packet path diversity we have implemented the proposed PPD-TCP schemes as part of the *ns-2* network simulator[1]. The first path that is used in our simulations is the default path determined by the routing algorithms available in *ns*. The second path is explicitly specified by means of a relay node. The relay node works bi-directionally and forwards TCP packets from the server to the client and vice versa. The PPD-sender node that wishes to send a packet along the second path simply addresses the relay server as the destination. The relay server receives and forwards the packet without any delay to the PPD receiver node.

Fig. 2 shows the first network topology under investigation. The first path follows the nodes N1 to N4 which is the default path, while the second path is along nodes N5 to N8 with N7 being the relay server. Each of these nodes has 20 data sources attached, with a large amount of TCP and UDP traffic heading for different destinations and competing for the network resources. We use the log-normal model proposed in [4] for the file size of each TCP session. UDP traffic is randomly switched on and off at each source. In our simulations we control the path load by varying the idle time between successive TCP and UDP transmissions. We measure the network path load as the ratio of average data throughput to the rate of the link. In Fig. 2 all links have 10Mbit/s transmission rate and introduce 20ms propagation delay.

[1] http://www.isi.edu/nsnam/ns/

5.1 Comparison of PPD-TCP variants

Fig. 3 shows the average time required to transmit a file of size 100Kbyte from the TCP server to the TCP client node in Fig. 2 as a function of network path load. The TCP data packet size is 1000Byte. The cross traffic is adjusted such that on average both paths have the same load. We record the time when a session starts and the time when the entire file has been successfully transmitted. The transmission times shown in Fig. 3 are averaged times over 1500 file transfers. In the following we name the different transport schemes as follows:

TCP	standard single-path TCP transport
PPD-TCP1	two-path TCP transport with duplicate ACKs (Section 3.1)
PPD-TCP2	two-path TCP transport with duplicate ACKs and duplicate data packets (Section 3.2)
PPD-TCP3	two-path TCP transport with duplicate ACKs and probe packets on the forward channel (Section 3.3)

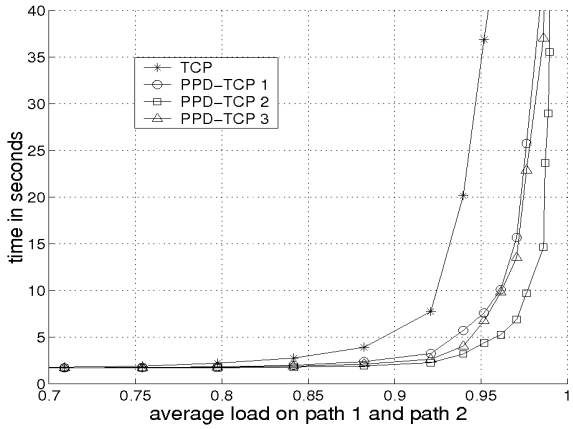


Figure 3: Average transmission time of a 100Kbyte media file versus network path load for the network topology in Fig. 2.

As the average network load increases, the time to transmit the file increases significantly for single-path TCP transport. As can be seen from Fig. 3 all three PPD-TCP variants significantly outperform single-path TCP transport for high path load. The largest gain, as expected, is obtained for the PPD-TCP scheme that duplicates both, data and acknowledgment packets. The performance of PPD-TCP1 and PPD-TCP3 for this simulation setup is very similar. This can be explained by the fact that both schemes exploit the advantage of having duplicate ACKs on the path from the client to the server. Congestion-dependent switching between the two

RED HAT 1012
RED HAT V. CAS

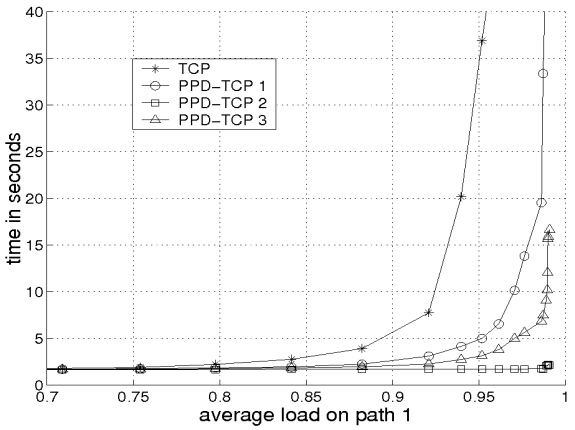


Figure 4: Transmission time versus network path load. Path 2 exhibits 20% lower average load than path 1.

forward paths, however, gives only a small advantage since both paths are equally loaded. The transmission time for our file at an average path load of 95% is reduced from around 35s for single-path TCP transport to about 7s for duplicated ACKs (PPD-TCP1), around 6s for duplicated acknowledgments in combination with probe packets (PPD-TCP3), and 4.5s for our upper bound scheme PPD-TCP2.

We expect even better performance of the PPD-TCP schemes if the two paths exhibit different average load. Fig. 4 shows the same simulation as in Fig 3 with the difference that the average load on Path 2 in Fig. 2 is 20% lower compared to Path 1. If we assume that the default path for single-path TCP transport corresponds to Path 1, we obtain the same curve as in Fig. 3. For all three PPD-TCP schemes, however, we see different results than in Fig. 3. For PPD-TCP 2, where data packets are duplicated, we observe a dramatic decrease in transmission time for the file at high loads on path 1. This is expected since the TCP server sees a much better connection (Path 2) to the client compared to the standard single-path TCP case. It is interesting to note in Fig. 4 that the use of probe packets (PPD-TCP3) turns out to be very beneficial. While the curves for PPD-TCP1 and PPD-TCP3 in Fig. 3 were almost identical, the PPD-TCP scheme with probe packets shows significant advantages here. This can be explained as follows. Even if we start sending the data packets along Path 1, the better transmission characteristics of Path 2 lead to an earlier arrival of the probe packets which then triggers switching of the active and the inactive data path. Therefore, most of the time, the data packets are transmitted along the path with the lower average load (Path 2) which leads to a performance improvement for our media file transmission.

For media delivery, the total transmission time of a media file is only one indicator of the improvements that are obtainable when combining TCP transport with packet path diversity. For continuous playout of media content at the receiver, uniform throughput is equally important. In the next experiment we investigate the throughput variation observed for the various PPD-TCP schemes in comparison to single-path TCP transmission. For this, we measure the size of the continuously playable media file at the receiver as a function of time. Only those packet numbers up to the first missing packed are counted since missing packets lead to interruption of playout due to the interdependency of successive packets.

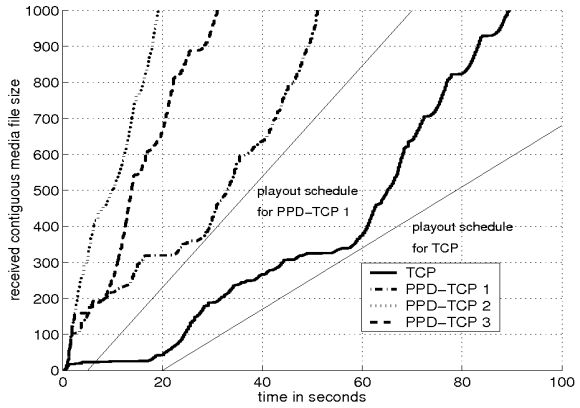


Figure 5: Playable media file size in packets at the receiver for average loads of 0.94 on Path1 and 0.94 on Path 2.

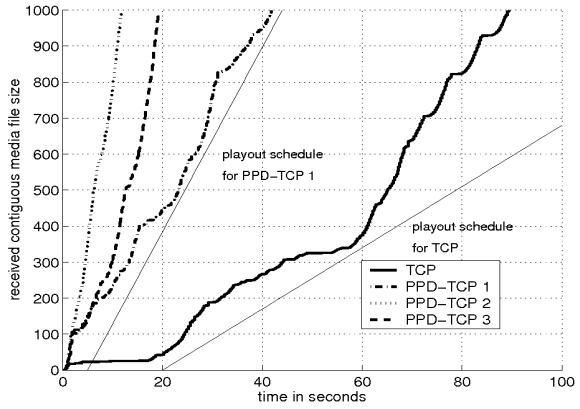


Figure 6: Playable media file size in packets at the receiver for average loads of 0.94 on path1 and 0.85 on Path 2.

Fig. 5 shows the playable file size as a function of time for identical load (94%) on Path 1 and Path 2. The total file size corresponds to 1000 TCP data packets each of size 1000 Byte. Single-path TCP performs very badly at this load level while the PPD-TCP variants lead to a significant improvement of the situation. Especially the probe packet based approach PPD-TCP3 comes very close to the duplicate data packet approach PPD-TCP2 which is very encouraging since the overhead introduced for this scheme is small. From the perspective of a media playout algorithm at the receiver, single-path TCP transmission requires a significant amount of pre-buffering in order to guarantee continuous playout during media file transfer. For the PPD-TCP schemes, the required pre-buffering is much smaller and therefore playout can start earlier and will be more robust with respect to receiver

RED HAT 1012
RED HAT V. CAS

buffer underflow. We show possible playout schedules that avoid playout interruption at the receiver for single-path TCP and PPD-TCP1 in Fig. 5 as thin solid lines. The received file size has to stay above that line in order to guarantee continuous playout. The point where this line intersects the x-axis represents the time we have to pre-buffer before playout can start and the slope of theses lines represent the maximum consumption rate our playout algorithm uses at the receiver. The steeper the line the higher this rate which translates into potentially higher quality of the media content. While 20 seconds pre-buffering is required for the single-path transmission only 5s pre-buffering is required for PPD-TCP1. Fig. 6 shows the same experiment as in Fig. 5 but for an average load level of 94% for Path 1 and 85% for Path 2. For this unbalanced case we observe a further significant improvement in throughput as can be seen for instance by the steeper playout deadline for PPD-TCP1.

So far we have considered packet path diversity examples where the two paths were completely independent. Now we investigate the performance of our proposed multi-path TCP transport schemes when the two paths share links. Naturally, shared links occur if we have to implement the PPD-functionality on the TCP host and if only a single link is available to access the network. Shared links will also be present if the two paths have to share a link due to a missing alternative (e.g., a cross-Atlantic link). In this case, however, the shared link can be assumed to be of high speed. It is therefore unlikely that this shared link represents the bottleneck link of the two paths that introduces significant correlation. Fig. 7 shows the topology that is used for the following experiments. The PPD-functionality is implemented on the TCP server host which has a single connection to the network. The two paths share a 20Mbit/s link between nodes N1 and N2 which introduces a small correlation between the two paths. The correlation is small since the bottleneck for each path are the 10Mbit/s links that are not shared. As before, we vary the TCP and UDP cross traffic to simulate different levels of network load. The number of TCP and UDP cross traffic sources attached to nodes N1 and N2 is now twice as large as before with half the cross traffic heading to and coming from Path 1. The other half heads towards and comes from Path 2. This way we can still run our simulations with equal average network load on both paths.

Fig. 8 shows the average transmission time of the media file as a function of network load for the shared link topology in Fig. 7. It can be seen from Fig. 8 that for the shared path example with low correlation we again observe a significant improvement for path diversified TCP transport.

When we increase the correlation between the two paths, we expect the gains for PPD-TCP to become smaller. This is illustrated in Fig. 9 where we repeat the previous experiment with the difference that the shared link now only has a link speed of 15Mbit/s. The number of TCP and UDP sources that produce cross traffic and their scheduling remain fixed. The link between nodes N1 and N2 now becomes the common bottleneck link

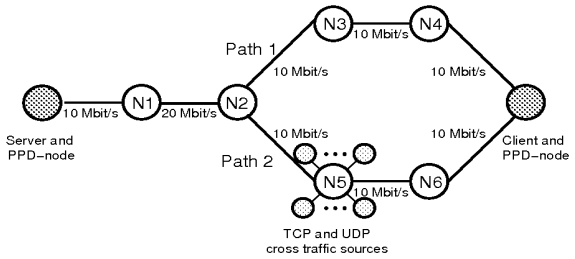


Figure 7: Multi-hop topology with shared link for *ns* network simulations. Each of the nodes N1 through N6 has a number of TCP data sources and UDP traffic sources attached (only shown for node N5).

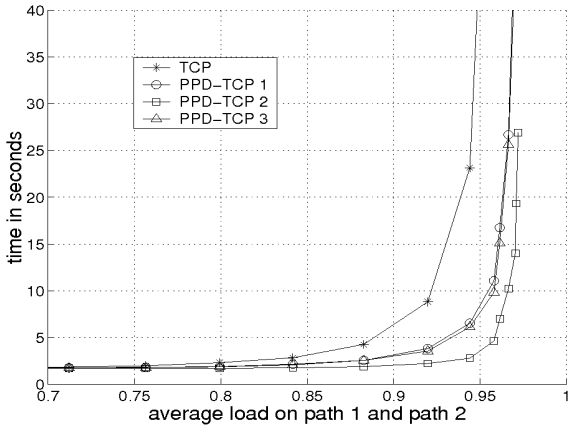


Figure 8: Average duration of a 100Kbyte media file transfer versus network path load. Both paths exhibit identical average network load. The load for Path 1 is measured between nodes N2 and N3 while the load for Path 2 is measured between nodes N2 and N5.

for the two paths. Fig. 9 shows the average transmission time of the 100Kbyte media file as a function of network load for this case. It can be seen from Fig. 9 that for the shared path example with high correlation we observe much smaller improvements for our PPD-TCP schemes.

5.2 Estimation of ΔT

In this section we provide simulation results for the measurement of the time difference ΔT that is used in Section 3.3 for determining the virtual arrival time of a data packet given that the corresponding probe packet has been received. In the first experiment we consider the symmetric network topology in Fig. 2 with all nodes being connected with 10Mbit/s links. We send a packet pair (probe packet first and data packet second) along the inactive path and measure ΔT for different network loads. Table 1 lists the average ΔT for different loads.

Since ΔT is mainly the difference of transmission times at zero network load, we can compute the time dif-

Table 1: Delay difference between the data and probe packet for the path with 10Mbit/s links.

Load	0	0.67	0.75	0.84	0.95	0.99
ΔT (ms)	4.6	3.6	3.5	3.4	3.3	3.1

RED HAT 1012
RED HAT V. CAS

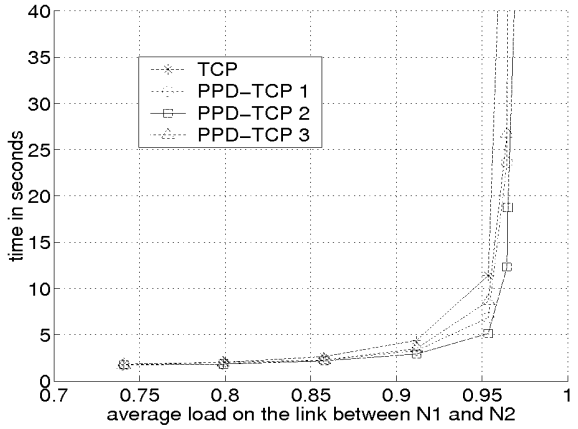


Figure 9: Average transmission time of a 100Kbyte media file versus network path load for the shared link topology in Fig. 7 with a 15Mbit/s link between nodes N1 and N2. The load is measured between nodes N1 and N2.

ference also analytically if we know the link transmission rates and the number of hops as

$$\Delta T = \frac{S_{data} - S_{probe}}{R_{link}} \times N_{hops} \quad (1)$$
$$\frac{(1000-40)\times 8bits}{10^7 bits/s} \times 6 = 4.6ms.$$

where S_{data} and S_{probe} are the packet sizes of the data and the probe packet, respectively. N_{hops} is the number of hops for the path (6 in Fig. 2) and R_{link} is the transmission rate on the links.

In practice we typically do not know the number of hops and the individual link rates and therefore have to measure ΔT regularly. At higher load, ΔT decreases since the first packet in a packet pair is more likely to be slowed down due to the traffic in front of it. However, this decrease is less than 2ms even at very high load.

Although the delay difference is very small for the path with 10Mbit/s links, it can be much larger for links with lower bandwidth where packets take longer to transmit. We also measure ΔT as a function of network load for a network topology where the second path in Fig. 2 has links of 1Mbit/s instead of 10Mbit/s. The measured results are listed in Table 2.

The analytical value for ΔT in (2) now becomes 46ms. Since ΔT is large in this simulation setup, the virtual delay of a data packet is much greater than the actual delay of the probe packet, which means we cannot directly compare the delay of the probe packet on the inactive path to that of the data packet on the active path. For example, if the end-to-end latency of the probe packets on the inactive path is much smaller due to its lower propagation delay, it does not necessarily mean that the inactive path is superior. If the transmission rate is very limited on that path, the delay a larger data packet would experience can be significantly higher.

Table 2: Delay difference between the data and probe packet for the path with 1Mbit/s links.

Load	0	0.68	0.77	0.85	0.94	0.99
ΔT (ms)	46.1	35.5	35.0	33.3	33.7	32.6

In practice, we use a threshold of twice ΔT in determining whether to switch paths or not. This alleviates the problem that ΔT is smaller than it should be at higher load. Therefore we avoid frequent switching between paths unless the monitored channel condition for the inactive path is significantly superior.

In the next experiment, we study how effective the estimation of the virtual delay of the data packet is using the packet pair technique. We compare the estimated virtual delay of the data packet to actually measured values. The latter are obtained by repeating the simulation under exactly the same conditions, but data packets are sent over the inactive path instead of the probe packets. The estimated and the actual value are plotted in Fig. 10 for the two different network topologies used before. The estimated and the actually measured end-to-end values agree closely.

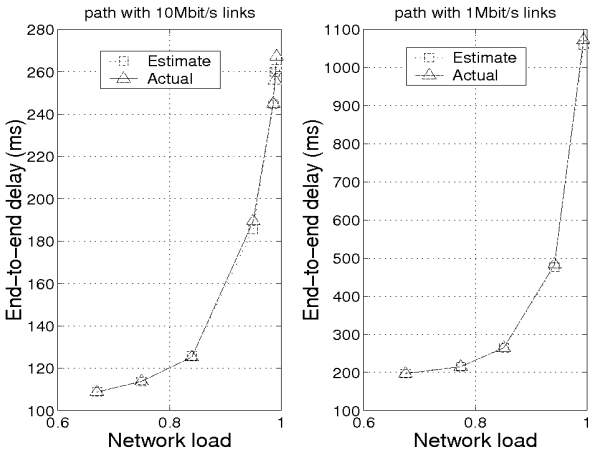


Figure 10: Estimated and actual end-to-end delays of data packets.

6 Conclusions

We study TCP-based file transfer and media transport in the Internet in combination with packet path diversity. The TCP protocol implementation remains unmodified in our study. We propose two different versions of packet path diversified TCP transport where in the first scheme only the acknowledgment packets from the TCP client to the sender are duplicated and sent along different paths. In the second scheme, probe packets are sent on the alternative path from the TCP server to the client in addition to the duplicated acknowledgment packets sent from the TCP client to the server. For this scheme we propose a simple strategy that is used to decide when to switch between the active and the inactive data path. This strategy is based on regularly measuring the arrival times of corresponding data and probe packets. Since the acknowledgment packets and probe packets are much smaller than the data packets, only a small overhead is introduced.

RED HAT 1012
RED HAT V. CAS

For comparison, we have implemented a third scheme where both, data and acknowledgment packets are path-diversified.

Our simulations show that the proposed PPD-TCP schemes lead to greatly improved time-critical file transfer and delivery of media content at high network load if the correlation between the two network paths is low. For the case of shared links that introduce correlation between the path congestion we still observe noticeable improvements. Only for a shared bottleneck link the diversity gains become small. For the maximum diversity gain it is important to select the two network paths such that the correlation between the two paths is minimized. This not necessarily means that the two paths should not have links in common but it is important that the bottleneck links are not shared by the two paths.

REFERENCES

[1] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "The case for resilient overlay networks," Proc. of the 8th Annual Workshop on Hot Topics in Operating Systems (HotOS-VIII), May 2001.

[2] J. G. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," SPIE Visual Communication and Image Processing, pp. 392-409, Jan. 2001.

[3] J. G. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On Multiple Description Streaming with Content Delivery Networks," IEEE INFOCOM '02, Jun. 2002.

[4] M. Arlitt and T. Jin, "Workload characterization study of the 1998 World Cup web site," IEEE Network, vol. 14, no. 3, pp. 30-37, May 2000.

[5] A. Banerjea, "Simulation study of the capacity effects of dispersity routing for fault-tolerant real-time channels," ACM SIGCOMM '96, vol. 26, no. 4, pp. 194-205, Oct. 1996.

[6] N. Gogate and S.S. Panwar, "Supporting video/image applications in a mobile multihop radio environment using route diversity," Proc. Int. Conf. Communications, pp. 833-844, San Jose, California, Jan., 1999.

[7] N. Gogate, D. Chung, S.S. Panwar, and Y. Wang, "Supporting image/video applications in a mobile multihop radio environment using route diversity and multiple description coding," IEEE Trans. on CSVT, to appear.

[8] E. Gustafsson and G. Karlsson, "A Literature Survey of Traffic Dispersion," IEEE Network Magazine, vol. 11, no. 2, pp. 28-36, March/April 1997.

[9] R. Krishnan and J.A. Silvester, "Choice of Allocation Granularity in Multipath Source Routing Schemes," IEEE INFOCOM '93, pp. 322-329, March 1993.

[10] Y. Liang, E. Steinbach, and B. Girod, "Real-time Voice Communication over the Internet Using Packet Path Diversity," ACM Multimedia '01, Ottawa, Canada, Oct. 2001.

[11] N.F. Maxemchuk, "Dispersity Routing in Store and Forward Networks," Ph.D. thesis, University of Pennsylvania, May 1975.

[12] Microsoft, Windows Media Player software and documentation
`http://www.windowsmedia.com`

[13] Real Networks, RealPlayer software and documentation
`http://www.realnetworks.com`

[14] W. R. Stevens, "TCP/IP Illustrated, Volume 1: The Protocols," Addison-Wesley, 1994, ISBN 0-201-63346-9.

[15] H. Suzuki and F.A. Tobagi, "Fast bandwidth reservation scheme with multi-link and multi-path routing in ATM networks", IEEE INFOCOM '92. pp. 2233-2240, 1992.

[16] Vivo Software, VivoActive software and documentation
`http://www.vivo.com`

RED HAT 1012
RED HAT V. CAS

Appeared in IEEE/ACM Transactions on Networking, Dec.1997. This is a much-extended and revised version of a paper that appeared at ACM SIGCOMM, 1996.

A Comparison of Mechanisms for Improving TCP Performance over Wireless Links

Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan and Randy H. Katz[1]
{hari,padmanab,ss,randy}@cs.berkeley.edu
Computer Science Division, Department of EECS, University of California at Berkeley

Abstract

Reliable transport protocols such as TCP are tuned to perform well in traditional networks where packet losses occur mostly because of congestion. However, networks with wireless and other lossy links also suffer from significant losses due to bit errors and handoffs. TCP responds to all losses by invoking congestion control and avoidance algorithms, resulting in degraded end-to-end performance in wireless and lossy systems. In this paper, we compare several schemes designed to improve the performance of TCP in such networks. We classify these schemes into three broad categories: end-to-end protocols, where loss recovery is performed by the sender; link-layer protocols, that provide local reliability; and split-connection protocols, that break the end-to-end connection into two parts at the base station. We present the results of several experiments performed in both LAN and WAN environments, using throughput and goodput as the metrics for comparison.

Our results show that a reliable link-layer protocol that is TCP-aware provides very good performance. Furthermore, it is possible to achieve good performance without splitting the end-to-end connection at the base station. We also demonstrate that selective acknowledgments and explicit loss notifications result in significant performance improvements.

1. Introduction

The increasing popularity of wireless networks indicates that wireless links will play an important role in future internetworks. Reliable transport protocols such as TCP [24, 26] have been tuned for traditional networks comprising wired links and stationary hosts. These protocols assume *congestion* in the network to be the primary cause for packet losses and unusual delays. TCP performs well over such networks by adapting to end-to-end delays and congestion losses. The TCP sender uses the cumulative acknowledgments it receives to determine which packets have reached the receiver, and provides reliability by retransmitting lost packets. For this purpose, it maintains a running average of the estimated round-trip delay and the mean linear deviation from it. The sender identifies the loss of a packet either by the arrival of several duplicate cumulative acknowledgments or the absence of an acknowledgment for the packet within a *timeout* interval equal to the sum of the smoothed round-trip delay and four times its mean deviation. TCP reacts to packet losses by dropping its transmission (congestion) window size before retransmitting packets, initiating congestion control or avoidance mechanisms (e.g., slow start [13]) and backing off its retransmission timer (Karn's Algorithm [16]). These measures result in a reduction in the load on the intermediate links, thereby controlling the congestion in the network.

Unfortunately, when packets are lost in networks for reasons other than congestion, these measures result in an unnecessary reduction in end-to-end throughput and hence, sub-optimal performance. Communication over wireless links is often characterized by sporadic high bit-error rates, and intermittent connectivity due to handoffs. TCP performance in such networks suffers from significant throughput degradation and very high interactive delays [8].

Recently, several schemes have been proposed to the alleviate the effects of non-congestion-related losses on TCP performance over networks that have wireless or similar high-loss links [3, 7, 28]. These schemes choose from a variety of mechanisms, such as local retransmissions, split-TCP connections, and forward error correction, to improve end-to-end throughput. However, it is unclear to what extent each of the mechanisms contributes to the improvement in performance. In this paper, we examine and compare the effectiveness of these schemes and their variants, and experimentally analyze the individual mechanisms and the degree of performance improvement due to each.

There are two different approaches to improving TCP performance in such lossy systems. The first approach hides any non-congestion-related losses from the TCP sender and therefore requires no changes to existing sender implementations. The intuition behind this approach is that since the problem is local, it should be solved locally, and that the transport layer need not be aware of the characteristics of the individual links. Protocols that adopt this approach attempt to make the lossy link appear as a higher quality link with a reduced effective bandwidth. As a result, most of the losses seen by the TCP sender are caused by congestion.

1. Web page URL http://daedalus.cs.berkeley.edu.
Srinivasan Seshan is now at IBM T.J. Watson Research Center, Hawthorne, NY (srini@watson.ibm.com).

RED HAT 1012
RED HAT V. CAS

Examples of this approach include wireless links with reliable link-layer protocols such as AIRMAIL [1], split connection approaches such as Indirect-TCP [3], and TCP-aware link-layer schemes such as the snoop protocol [7]. The second class of techniques attempts to make the sender aware of the existence of wireless hops and realize that some packet losses are not due to congestion. The sender can then avoid invoking congestion control algorithms when non-congestion-related losses occur — we describe some of these techniques in Section 3. Finally, it is possible for a wireless-aware transport protocol to coexist with link-layer schemes to achieve good performance.

We classify the many schemes into three basic groups, based on their fundamental philosophy: end-to-end proposals, split-connection proposals and link-layer proposals. The end-to-end protocols attempt to make the TCP sender handle losses through the use of two techniques. First, they use some form of selective acknowledgments (SACKs) to allow the sender to recover from multiple packet losses in a window without resorting to a coarse timeout. Second, they attempt to have the sender distinguish between congestion and other forms of losses using an Explicit Loss Notification (ELN) mechanism. At the other end of the solution spectrum, split-connection approaches completely hide the wireless link from the sender by terminating the TCP connection at the base station. Such schemes use a separate reliable connection between the base station and the destination host. The second connection can use techniques such as negative or selective acknowledgments, rather than just standard TCP, to perform well over the wireless link. The third class of protocols, link-layer solutions, lie between the other two classes. These protocols attempt to hide link-related losses from the TCP sender by using local retransmissions and perhaps forward error correction [e.g., 18] over the wireless link. The local retransmissions use techniques that are tuned to the characteristics of the wireless link to provide a significant increase in performance. Since the end-to-end TCP connection passes through the lossy link, the TCP sender may not be fully shielded from wireless losses. This can happen either because of timer interactions between the two layers [10], or more likely because of TCP's duplicate acknowledgments causing sender fast retransmissions even for segments that are locally retransmitted. As a result, some proposals to improve TCP performance use mechanisms based on the knowledge of TCP messaging to shield the TCP sender more effectively and avoid competing and redundant retransmissions [7].

In this paper, we evaluate the performance of several end-to-end, split-connection and link-layer protocols using end-to-end throughput and goodput as performance metrics, in both LAN and WAN configurations. In particular, we seek to answer the following specific questions:

1. What combination of mechanisms results in best performance for each of the protocol classes?
2. How important is it for link-layer schemes to be aware of TCP algorithms to achieve high end-to-end throughput?
3. How useful are selective acknowledgments in dealing with lossy links, especially in the presence of burst losses?
4. Is it important for the end-to-end connection to be split in order to effectively shield the sender from wireless losses and obtain the best performance?

We answer these questions by implementing and testing the various protocols in a wireless testbed consisting of Pentium PC base stations and IBM ThinkPad mobile hosts communicating over a 915 MHz AT&T Wavelan, all running BSD/OS 2.0. For each protocol, we measure the end-to-end throughput, and goodputs for the wired and (one-hop) wireless paths. For any path (or link), goodput is defined as the ratio of the actual transfer size to the total number of bytes transmitted over that path. In general, the wired and wireless goodputs differ because of wireless losses, local retransmissions and congestion losses in the wired network. These metrics allow us to determine the end-to-end performance as well as the transmission efficiency across the network. While we used a wireless hop as the lossy link in our experiments, we believe our results are applicable in a wider context to links where significant losses occur for reasons other than congestion. Examples of such links include high-speed modems and cable modems.

We show that a reliable link-layer protocol with some knowledge of TCP results in very good performance. Our experiments indicate that shielding the TCP sender from duplicate acknowledgments caused by wireless losses improves throughput by 10-30%. Furthermore, it is possible to achieve good performance without splitting the end-to-end connection at the base station. We also demonstrate that selective acknowledgments and explicit loss notifications result in significant performance improvements. For instance, the simple ELN scheme we evaluated improved the end-to-end throughput by a factor of more than two compared to TCP Reno, with comparable goodput values.

The rest of this paper is organized as follows. Section 2 briefly describes some proposed solutions to the problem of reliable transport protocols over wireless links. Section 3 describes the implementation details of the different protocols in our wireless testbed, and Section 4 presents the results and analysis of several experiments. Section 5 discusses some miscellaneous issues related to handoffs, ELN implementation and selective acknowledgments. We present our conclusions in Section 6, and mention some future work in Section 7.

RED HAT 1012
RED HAT V. CAS

2. Related Work

In this section, we summarize some protocols that have been proposed to improve the performance of TCP over wireless links. We also briefly describe some proposed methods to add SACKs to TCP.

- **Link-layer protocols**: There have been several proposals for reliable link-layer protocols. The two main classes of techniques employed by these protocols are: error correction, using techniques such as forward error correction (FEC), and retransmission of lost packets in response to automatic repeat request (ARQ) messages. The link-layer protocols for the digital cellular systems in the U.S. — both CDMA [15] and TDMA [22] — primarily use ARQ techniques. While the TDMA protocol guarantees reliable, in-order delivery of link-layer frames, the CDMA protocol only makes a limited attempt and leaves eventual error recovery to the (reliable) transport layer. Other protocols like the AIRMAIL protocol [1] employ a combination of FEC and ARQ techniques for loss recovery.

  The main advantage of employing a link-layer protocol for loss recovery is that it fits naturally into the layered structure of network protocols. The link-layer protocol operates independently of higher-layer protocols and does not maintain any per-connection state. The main concern about link-layer protocols is the possibility of adverse effect on certain transport-layer protocols such as TCP, as described in Section 1. We investigate this in detail in our experiments.

- **Split connection protocols [3, 28]:** Split connection protocols split each TCP connection between a sender and receiver into two separate connections at the base station — one TCP connection between the sender and the base station, and the other between the base station and the receiver. Over the wireless hop, a specialized protocol tuned to the wireless environment may be used. In [28], the authors propose two protocols — one in which the wireless hop uses TCP, and another in which the wireless hop uses a selective repeat protocol (SRP) on top of UDP. They study the impact of handoffs on performance and conclude that they obtain no significant advantage by using SRP instead of TCP over the wireless connection in their experiments. However, our experiments demonstrate benefits in using a simple selective acknowledgment scheme with TCP over the wireless connection.

  Indirect-TCP [Bakre95] is a split-connection solution that uses standard TCP for its connection over the wireless link. Like other split-connection proposals, it attempts to separate loss recovery over the wireless link from that across the wireline network, thereby shielding the original TCP sender from the wireless link. However, as our experiments indicate, the choice of TCP over the wireless link results in several performance problems. Since TCP is not well-tuned for the lossy link, the TCP sender of the wireless connection often times out, causing the original sender to stall. In addition, every packet incurs the overhead of going through TCP protocol processing twice at the base station (as compared to zero times for a non-split-connection approach), although extra copies are avoided by an efficient kernel implementation. Another disadvantage of split connections is that the end-to-end semantics of TCP acknowledgments is violated, since acknowledgments to packets can now reach the source even before the packets actually reach the mobile host. Also, since split-connection protocols maintain a significant amount of state at the base station per TCP connection, handoff procedures tend to be complicated and slow. Section 5.1 discusses some issues related to cellular handoffs and TCP performance.

- **The Snoop Protocol [7]:** The snoop protocol introduces a module, called the *snoop agent*, at the base station. The agent monitors every packet that passes through the TCP connection in both directions and maintains a cache of TCP segments sent across the link that have not yet been acknowledged by the receiver. A packet loss is detected by the arrival of a small number of duplicate acknowledgments from the receiver or by a local timeout. The snoop agent retransmits the lost packet if it has it cached and suppresses the duplicate acknowledgments. In our classification of the protocols, the snoop protocol is a link-layer protocol that takes advantage of the knowledge of the higher-layer transport protocol (TCP).

  The main advantage of this approach is that it suppresses duplicate acknowledgments for TCP segments lost and retransmitted locally, thereby avoiding unnecessary fast retransmissions and congestion control invocations by the sender. The per-connection state maintained by the snoop agent at the base station is *soft*, and is not essential for correctness. Like other link-layer solutions, the snoop approach could also suffer from not being able to completely shield the sender from wireless losses.

- **Selective Acknowledgments**: Since standard TCP uses a cumulative acknowledgment scheme, it often does not provide the sender with sufficient information to recover quickly from multiple packet losses within a single transmission window. Several studies [e.g., 11] have shown that TCP enhanced with selective acknowledgments performs better than standard TCP in such situations. SACKs were added as an option to TCP by RFC 1072 [14]. However, disagreements over the use of SACKs prevented the specification from being adopted, and the SACK option was removed from later TCP RFCs. Recently, there has been renewed interest in adding SACKs to TCP. Two relevant proposals are the

RED HAT 1012
RED HAT V. CAS

Name	Category	Special Mechanisms
E2E	end-to-end	standard TCP-Reno
E2E-NEWRENO	end-to-end	TCP-NewReno
E2E-SMART	end-to-end	SMART-based selective acks
E2E-IETF-SACK	end-to-end	IETF selective acks
E2E-ELN	end-to-end	Explicit Loss Notification (ELN)
E2E-ELN-RXMT	end-to-end	ELN with retransmit on first dupack
LL	link-layer	none
LL-TCP-AWARE	link-layer	duplicate ack suppression
LL-SMART	link-layer	SMART-based selective acks
LL-SMART-TCP-AWARE	link-layer	SMART and duplicate ack suppression
SPLIT	split-connection	none
SPLIT-SMART	split-connection	SMART-based wireless connection

Table 1. Summary of protocols studied in this paper.

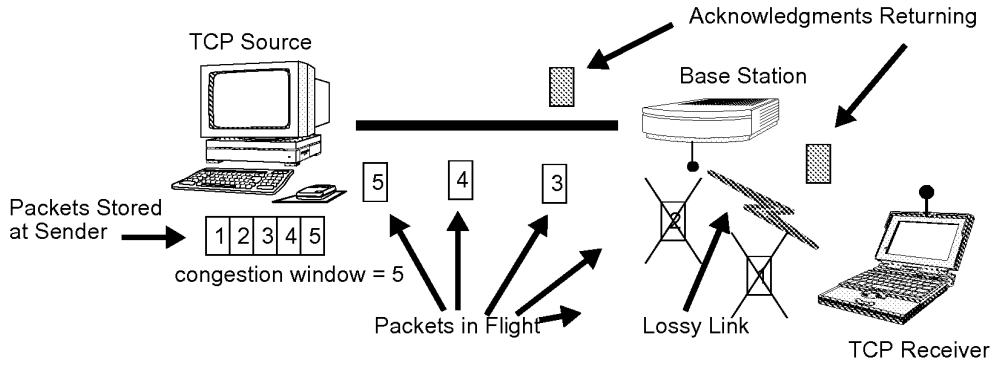


Figure 1. A typical loss situation

recent RFC on TCP SACKs [19] and the SMART scheme [17].

The SACK RFC proposes that each acknowledgment contain information about up to three non-contiguous blocks of data that have been received successfully by the receiver. Each block of data is described by its starting and ending sequence number. Due to the limited number of blocks, it is best to inform the sender about the most recent blocks received. The RFC does not specify the sender behavior, except to require that standard TCP congestion control actions be performed when losses occur.

An alternate proposal, SMART, uses acknowledgments that contain the cumulative acknowledgment and the sequence number of the packet that caused the receiver to generate the acknowledgment (this information is a subset of the three-blocks scheme proposed in the RFC). The sender uses this information to create a bitmask of packets that have been delivered successfully to the receiver. When the sender detects a gap in the bitmask, it immediately assumes that the missing packets have been lost without considering the possibility that they simply may have been reordered. Thus this scheme trades off some resilience to reordering and lost acknowledgments in exchange for a reduction in overhead to generate and transmit acknowledgments.

3. Implementation Details

This section describes the protocols we have implemented and evaluated. Table 1 summarizes the key ideas in each scheme and the main differences between them. Figure 1 shows a typical loss situation over the wireless link. Here, the TCP sender is in the middle of a transfer across a two-hop network to a mobile host. At the depicted time, the sender's congestion window consists of 5 packets. Of the five packets in the network, the first two packets are lost on the wireless link. As described in the rest of this section, each protocol reacts to these losses in different ways and

RED HAT 1012
RED HAT V. CAS

generates messages that result in loss recovery. Although this figure only shows data packets being lost, our experiments have wireless errors in both directions.

3.1 End-To-End Schemes

Although a wide variety of TCP versions are used on the Internet, the current de facto standard for TCP implementations is TCP Reno [26]. We call this the E2E protocol, and use it as the standard basis for performance comparison.

The E2E-NEWRENO protocol improves the performance of TCP-Reno after multiple packet losses in a window by remaining in fast recovery mode if the first new acknowledgment received after a fast retransmission is "partial", i.e, is less than the value of the last byte transmitted when the fast retransmission was done. Such partial acknowledgements are indicative of multiple packet losses within the original window of data. Remaining in fast recovery mode enables the connection to recover from losses at the rate of one segment per round trip time, rather than stall until a coarse timeout as TCP-Reno often would [9, 12].

The E2E-SMART and E2E-IETF-SACK protocols add SMART-based and IETF selective acknowledgments respectively to the standard TCP Reno stack. This allows the sender to handle multiple losses within a window of outstanding data more efficiently. However, the sender still assumes that losses are a result of congestion and invokes congestion control procedures, shrinking its congestion window size. This allows us to identify what percentage of the end-to-end performance degradation is associated with standard TCP's handling of error detection and retransmission. We used the SMART-based scheme [17] only for the LAN experiments. This scheme is well-suited to situations where there is little reordering of packets, which is true for one-hop wireless systems such as ours. Unlike the scheme proposed in [17], we do not use any special techniques to detect the loss of a retransmission. The sender retransmits a packet when it receives a SMART acknowledgment only if the same packet was not retransmitted within the last round-trip time. If no further SMART acknowledgments arrive, the sender falls back to the coarse timeout mechanism to recover from the loss. We used the IETF selective acknowledgement scheme both for the LAN and the WAN experiments. Our implementation is based on the RFC and takes appropriate congestion control actions upon receiving SACK information [4].

The E2E-ELN protocol adds an Explicit Loss Notification (ELN) option to TCP acknowledgments. When a packet is dropped on the wireless link, future cumulative acknowledgments corresponding to the lost packet are marked to identify that a non-congestion related loss has occurred. Upon receiving this information with duplicate acknowledgments, the sender may perform retransmissions without invoking the associated congestion-control procedures. This option allows us to identify what percentage of the end-to-end performance degradation is associated with TCP's incorrect invocation of congestion control algorithms when it does a fast retransmission of a packet lost on the wireless hop. The E2E-ELN-RXMT protocol is an enhancement of the previous one, where the sender retransmits the packet on receiving the first duplicate acknowledgement with the ELN option set (as opposed to the third duplicate acknowledgement in the case of TCP Reno), in addition to not shrinking its window size in response to wireless losses.

In practice, it might be difficult to identify which packets are lost due to errors on a lossy link. However, in our experiments we assume sufficient knowledge at the receiver about wireless losses to generate ELN information. We describe some possible implementation policies and strategies for the ELN mechanism in Section 5.2.

3.2 Link-Layer Schemes

Unlike TCP for the transport layer, there is no de facto standard for link-layer protocols. Existing link-layer protocols choose from techniques such as Stop-and-Wait, Go-Back-N, Selective Repeat and Forward Error Correction to provide reliability. Our base link-layer algorithm, called LL, uses cumulative acknowledgments to determine lost packets that are retransmitted locally from the base station to the mobile host. To minimize overhead, our implementation of LL leverages off TCP acknowledgments instead of generating its own. Timeout-based retransmissions are done by maintaining a smoothed round-trip time estimate, with a minimum timeout granularity of 200 ms to limit the overhead of processing timer events. This still allows the LL scheme to retransmit packets several times before a typical TCP Reno transmitter would time out. LL is equivalent to the snoop agent that does not suppress any duplicate acknowledgments, and does not attempt in-order delivery of packets across the link (unlike protocols proposed in [15], [22]).

While the use of TCP acknowledgments by our LL protocol renders it atypical of traditional ARQ protocols, we believe that it still preserves the key feature of such protocols: the ability to retransmit packets locally, independently of and on a much faster time scale than TCP. Therefore, we expect the qualitative aspects of our results to be applicable to general link-layer protocols.

We also investigated a more sophisticated link-layer protocol (LL-SMART) that uses selective retransmissions to improve performance. The LL-SMART protocol performs this by applying a SMART-based acknowledgment scheme at the link layer. Like the LL protocol, LL-SMART uses TCP acknowledgments instead of generating its own and limits its minimum timeout to 200 ms. LL-SMART is equivalent to the snoop agent performing retransmissions based on selective acknowledgements but not suppressing duplicate acknowledgments at the base station.

RED HAT 1012
RED HAT V. CAS

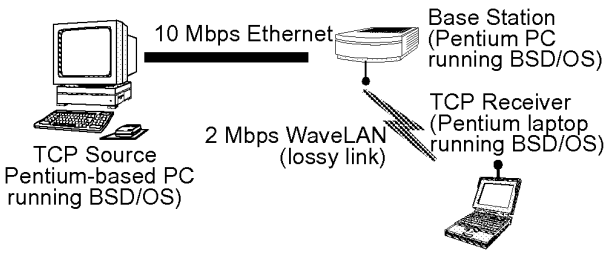


Figure 2. Experimental topology. There were an additional 16 Internet hops between the source and base station during the WAN experiments.

We added TCP awareness to both the LL and LL-SMART protocols, resulting in the LL-TCP-AWARE and LL-SMART-TCP-AWARE schemes. The LL-TCP-AWARE protocol is identical to the snoop protocol, while the LL-SMART-TCP-AWARE protocol uses SMART-based techniques for further optimization using selective repeat. LL-SMART-TCP-AWARE is the best link-layer protocol in our experiments — it performs local retransmissions based on selective acknowledgments and shields the sender from duplicate acknowledgments caused by wireless losses.

3.3 Split-Connection Schemes

Like I-TCP, our SPLIT scheme uses an intermediate host to divide a TCP connection into two separate TCP connections. The implementation avoids data copying in the intermediate host by passing the pointers to the same buffer between the two TCP connections. A variant of the SPLIT approach we investigated, SPLIT-SMART, uses a SMART-based selective acknowledgment scheme on the wireless connection to perform selective retransmissions. There is little chance of reordering of packets over the wireless connection since the intermediate host is only one hop away from the final destination.

4. Experimental Results

In this section, we describe the experiments we performed and the results we obtained, including detailed explanations for observed performance. We start by describing the experimental testbed and methodology. We then describe the performance of the various link-layer, end-to-end and split-connection schemes.

4.1 Experimental Methodology

We performed several experiments to determine the performance and efficiency of each of the protocols. The protocols were implemented as a set of modifications to the BSD/OS TCP/IP (Reno) network stack. To ensure a fair basis for comparison, none of the protocols implementations introduce any additional data copying at intermediate points from sender to receiver.

Our experimental testbed consists of IBM ThinkPad laptops and Pentium-based personal computers running BSD/OS 2.1 from BSDI. The machines are interconnected using a 10 Mbps Ethernet and 915 MHz AT&T WaveLANs [27], a shared-medium wireless LAN with a raw signalling bandwidth of 2 Mbps. The network topology for our experiments is shown in Figure 2. The peak throughput for TCP bulk transfers is 1.5 Mbps in the local area testbed and 1.35 Mbps in the wide area testbed in the absence of congestion or wireless losses. These testbed topologies represent typical scenarios of wireless links and mobile hosts, such as cellular wireless networks. In addition, our experiments focus on data transfer to the mobile host, which is the common case for mobile applications (e.g., Web accesses).

In order to measure the performance of the protocols under controlled conditions, we generate errors on the lossy link using an exponentially distributed bit-error model. The receiving entity on the lossy link generates an exponential distribution for each bit-error rate and changes the TCP checksum of the packet if the error generator determines that the packet should be dropped. Losses are generated in both directions of the wireless channel, so TCP acknowledgments are dropped too. The TCP data packet size in our experiments is 1400 bytes. We first measure and analyze the performance of the various protocols at an average error rate of one every 64 KBytes (this corresponds to a bit-error rate of about 1.9×10^{-6}). Note that since the exponential distribution has a standard deviation equal to its mean, there are several occasions when multiple packets are lost in close succession. We then report the results of some burst error situations, where between two and six packets are dropped in every burst (Section 4.5). Finally, we investigate the performance of many of these protocols across a range of error rates from one every 16 KB to one every 256 KB. The choice of the exponentially distributed error model is motivated by our desire to understand the precise dynamics of each protocol in response to a wireless loss, and is not an attempt to empirically model a wireless channel. While the actual performance numbers will be a function of the exact error model, the relative performance is dependent on how the protocol behaves after one or more losses in a single TCP window. Thus, we expect our overall conclusions to be applicable under other patterns of wireless loss as well. Finally, we believe that though wireless errors are generated artificially in our experiments, the use of a real testbed is still valuable in that it introduces realistic effects such as wireless bandwidth limitation, media access contention, protocol processing delays, etc., which are hard to model realistically in a simulation.

In our experiments, we attempt to ensure that losses are only due to wireless errors (and not congestion). This allows us to focus on the effectiveness of the mechanisms in handling such losses. The WAN experiments are performed across 16

RED HAT 1012
RED HAT V. CAS

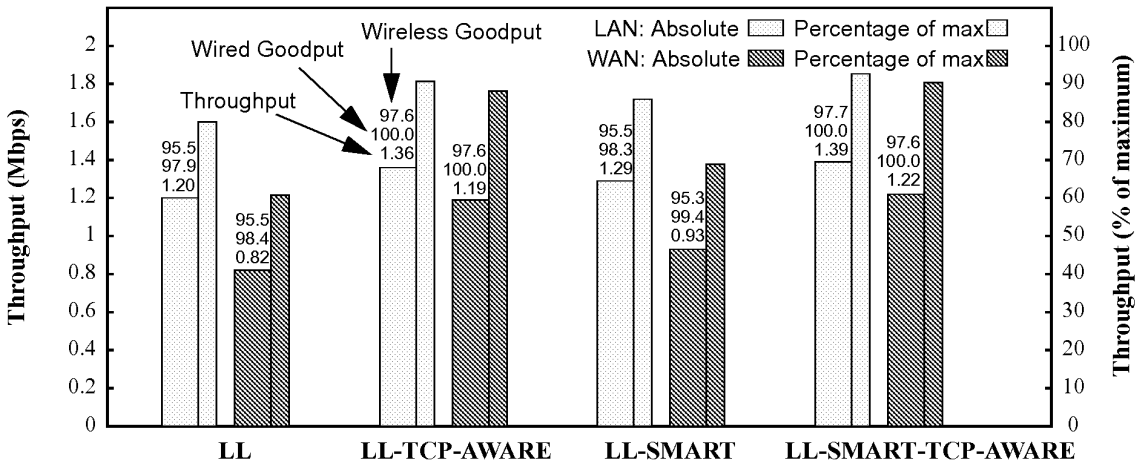


Figure 3. Performance of link-layer protocols: bit-error rate = 1.9×10^{-6} (1 error/65536 bytes), socket buffer size = 32 KB. For each case there are two bars: the thick one corresponds to the scale on the left and denotes the throughput in Mbps; the thin one corresponds to the scale on the right and shows the throughput as a percentage of the maximum, i.e. in the absence of wireless errors (1.5 Mbps in the LAN environment and 1.35 Mbps in the WAN environment).

Internet hops with minimal congestion[2] in order to study the impact of large delay-bandwidth products.

Each run in the experiment consists of an 8 MByte transfer from the source to receiver across the wired net and the WaveLAN link. We chose this rather long transfer size in order to limit the impact of transient behavior at the start of a TCP connection. During each run, we measure the throughput at the receiver in Mbps, and the wired and wireless goodputs as percentages. In addition, all packet transmissions on the Ethernet and WaveLan are recorded for analysis using tcpdump [20], and the sender's TCP code instrumented to record events such as coarse timeouts, retransmission times, duplicate acknowledgment arrivals, congestion window size changes, etc. The rest of this section presents and discusses the results of these experiments.

4.2 Link-Layer Protocols

Traditional link-layer protocols operate independently of the higher-layer protocol, and consequently, do not necessarily shield the sender from the lossy link. In spite of local retransmissions, TCP performance could be poor for two reasons: (i) competing retransmissions caused by an incompatible setting of timers at the two layers, and (ii) unnecessary invocations of the TCP fast retransmission mechanism due to out-of-order delivery of data. In [10], the effects of the first situation are simulated and analyzed for a TCP-like transport protocol (that *closely* tracks the round-trip time to set its retransmission timeout) and a reliable link-layer protocol. The conclusion was that unless the packet loss rate is high (more than about 10%), competing retransmissions by the link and transport layers often lead to significant performance degradation. However, this is not the dominating effect when link layer schemes, such as LL, are used with TCP Reno and its variants. These TCP implementations have coarse retransmission timeout granularities that are typically multiples of 500 ms, while link-layer protocols typically have much finer timeout granularities. The real problem is that when packets are lost, link-layer protocols that do not attempt in-order delivery across the link (e.g., LL) cause packets to reach the TCP receiver out-of-order. This leads to the generation of duplicate acknowledgments by the TCP receiver, which causes the sender to invoke fast retransmission and recovery. This can potentially cause degraded throughput and goodput, especially when the delay-bandwidth product is large.

Our results substantiate this claim, as can be seen by comparing the LL and LL-TCP-AWARE results (Figure 3 and Table 2). For a packet size of 1400 bytes, a bit error rate of 1.9×10^{-6} (1/65536 bytes) translates to a packet error rate of about 2.2 to 2.3%. Therefore, an optimal link-layer protocol that recovers from errors locally and does not compete with TCP retransmissions should have a wireless goodput of 97.7% and a wired goodput of 100% in the absence of congestion. In the LAN experiments, the throughput difference between LL and LL-TCP-AWARE is about 10%. However, the LL wireless goodput is only 95.5%, significantly less than LL-TCP-AWARE's wireless goodput of 97.6%, which is close to the maximum achievable goodput. When a loss occurs, the LL protocol performs a local retransmission relatively quickly. However, enough packets are typically in transit to create more than 3 duplicate acknowledgments. These duplicates eventually propagate to the sender and trigger a fast retransmission and the associated congestion control mechanisms. These fast retransmissions result in

2. WAN experiments across the US were performed between 10 pm and 4 am, PST and we verified that no congestion losses occurred in the runs reported.

RED HAT 1012
RED HAT V. CAS

	LL	LL-TCP-AWARE	LL-SMART	LL-SMART-TCP-AWARE
LAN (8 KB)	1.20 (95.6%,97.9%)	1.29 (97.6%,100%)	1.29 (96.1%,98.9%)	1.37 (97.6%,100%)
LAN (32 KB)	1.20 (95.5%,97.9%)	1.36 (97.6%,100%)	1.29 (95.5%,98.3%)	1.39 (97.7%,100%)
WAN (32 KB)	0.82 (95.5%,98.4%)	1.19 (97.6%,100%)	0.93 (95.3%,99.4%)	1.22 (97.6%,100%)

Table 2. This table summarizes the results for the link-layer schemes for an average error rate of one every 65536 bytes of data. Each entry is of the form: throughput (wireless goodput, wired goodput). Throughput is measured in Mbps. Goodput is expressed as a percentage.

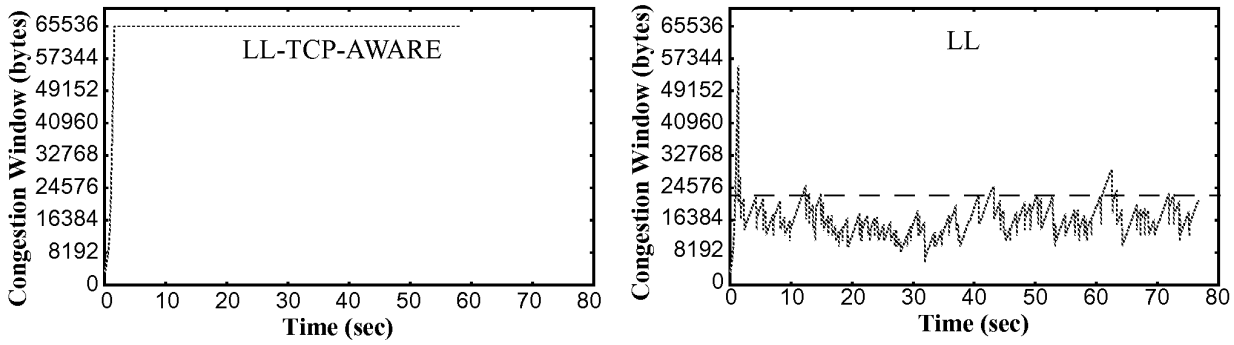


Figure 4. Congestion window size for link-layer protocols in wide area tests. The horizontal dashed line in the LL graph shows the 23000 byte WAN bandwidth-delay product.

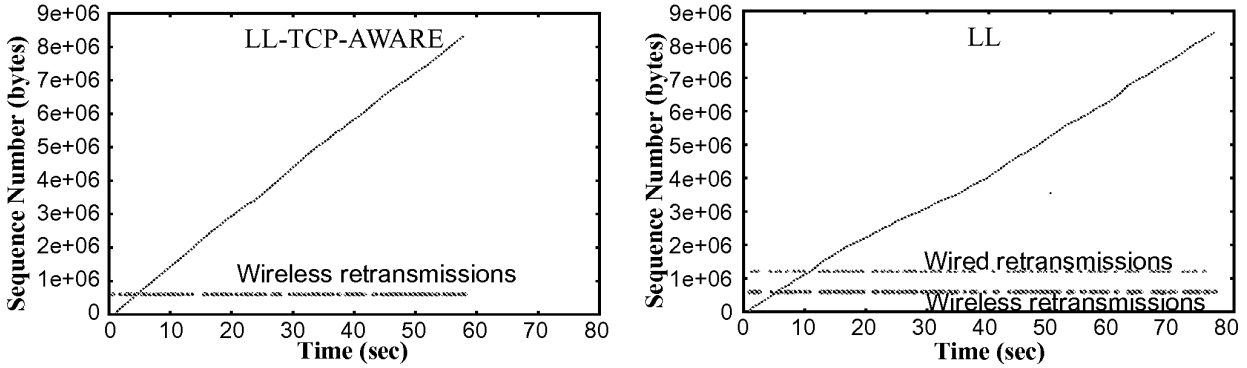


Figure 5. Packet sequence traces for LL-TCP-AWARE and LL. No coarse timeouts occur in either case. For LL-TCP-AWARE, the horizontal row of dots shows the times of wireless link retransmissions. For LL, the top row shows sender fast retransmission times and the bottom row shows both local wireless and sender retransmissions.

reduced goodput; about 90% of the lost packets are retransmitted by both the source and the base station.

The effects of this interaction are much more pronounced in the wide-area experiments — the throughput difference is about 30% in this case. The cause for the more pronounced deterioration in performance is the higher bandwidth-delay product of the wide-area connection. The LL scheme causes the sender to invoke congestion control procedures often due to duplicate acknowledgments and causes the average window size of the transmitter to be lower than for LL-TCP-AWARE. This is shown in Figure 4, which compares the congestion window size of LL and LL-TCP-AWARE as a function of time. Note that the number of outstanding data bytes in the network is the minimum of the congestion window and the receiver advertised window. This is bounded by the receiver's socket buffer size. In the congestion window graphs for each protocol, the receiver socket buffer is 32KB.

In the wide area, the bandwidth-delay product is about 23000 bytes (1.35 Mbps * 135 ms), and the congestion window drops below this value several times during each TCP transfer. On the other hand, the LAN experiments do not suffer from such a large throughput degradation because LL's lower congestion-window size is usually still larger than the connection's delay-bandwidth product of about 1900 bytes (1.5 Mbps * 10 ms). Therefore, the LL scheme can maintain a nearly full "data pipe" between the sender and receiver in the local connection but not in the wide area

RED HAT 1012
RED HAT V. CAS

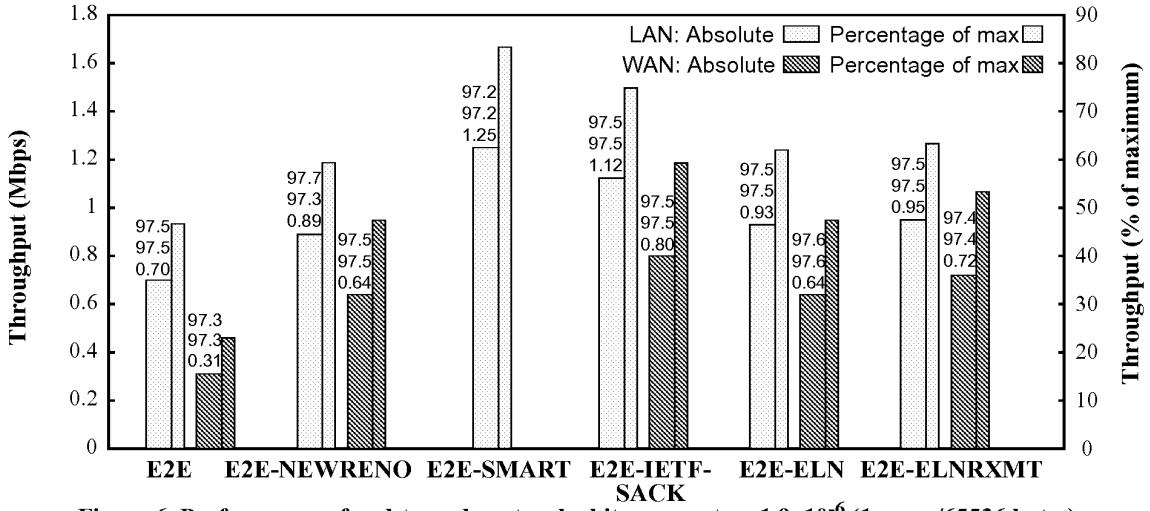


Figure 6. Performance of end-to-end protocols: bit error rate = 1.9×10^{-6} (1 error/65536 bytes).

	E2E	E2E-NEWRENO	E2E-SMART	E2E-IETF-SACK	E2E-ELN	E2E-ELN-RXMT
LAN (8 KB)	0.55 (97.0,96.0)	0.66 (97.3,97.3)	1.12 (97.6,97.6)	0.68 (97.3,97.3)	0.69 (97.3,97.2)	0.86 (97.4,97.3)
LAN (32 KB)	0.70 (97.5,97.5)	0.89 (97.7,97.3)	1.25 (97.2,97.2)	1.12 (97.5,97.5)	0.93 (97.5,97.5)	0.95 (97.5,97.5)
WAN (32 KB)	0.31 (97.3,97.3)	0.64 (97.5,97.5)	N.A.	0.80 (97.5,97.5)	0.64 (97.6,97.6)	0.72 (97.4,97.4)

Table 3. This table summarizes the results for the end-to-end schemes for an average error rate of one every 65536 bytes of data. The numbers in the cells follow the same convention as in Table 2.

one. The 10% LAN degradation is almost entirely due to the excessive retransmissions over the wireless link and to the smaller average congestion window size compared to LL-TCP-AWARE. Another important point to note is that LL successfully prevents coarse timeouts from happening at the source. Figure 5 shows the sequence traces of TCP transfers for LL-TCP-AWARE and LL.

In summary, our results indicate that a simple link-layer retransmission scheme does not entirely avoid the adverse effects of TCP fast retransmissions and the consequent performance degradation. An enhanced link-layer scheme that uses knowledge of TCP semantics to prevent duplicate acknowledgments caused by wireless losses from reaching the sender and locally retransmits packets achieves significantly better performance.

4.3 End-To-End Protocols

The performance of the various end-to-end protocols is summarized in Figure 6 and Table 3. The performance of TCP Reno, the baseline E2E protocol, highlights the problems with TCP over lossy links. At a 2.3% packet loss rate (as explained in Section 4.2), the E2E protocol achieves a throughput of less than 50% of the maximum (i.e., throughput in the absence of wireless losses) in the local-area and less than 25% of the maximum in the wide-area experiments. However, all the end-to-end protocols achieve goodputs close to the optimal value of 97.7%. The primary reason for the low throughput is the large number of timeouts that occur during the transfer (Figure 7). The resulting average window size during the transfer is small, preventing the "data pipe" from being kept full and reducing the effectiveness of the fast retransmission mechanism (Figure 8).

The modified end-to-end protocols improve throughput by retransmitting packets known to have been lost on the wireless hop earlier than they would have been by the baseline E2E protocol, and by reducing the fluctuations in window size. The E2E-NEWRENO, E2E-ELN, E2E-SMART and E2E-IETF-SACK protocols each use new TCP options and more sophisticated acknowledgment processing techniques to improve the speed and accuracy of identifying and retransmitting lost packets, as well as by recovering from multiple losses in a single transmission window without timing out. The remainder of this section discusses the benefits of three techniques — partial acknowledgments, explicit loss notifications, and selective acknowledgments.

Partial acknowledgments: E2E-NEWRENO, which uses partial acknowledgment information to recover from multiple losses in a window at the rate of one packet per round-trip time, performs between 10 and 25% better than E2E over a LAN and about 2 times better than E2E in the WAN experiments. The performance improvement is a function of the socket buffer size — the larger the buffer size, the better

RED HAT 1012
RED HAT V. CAS

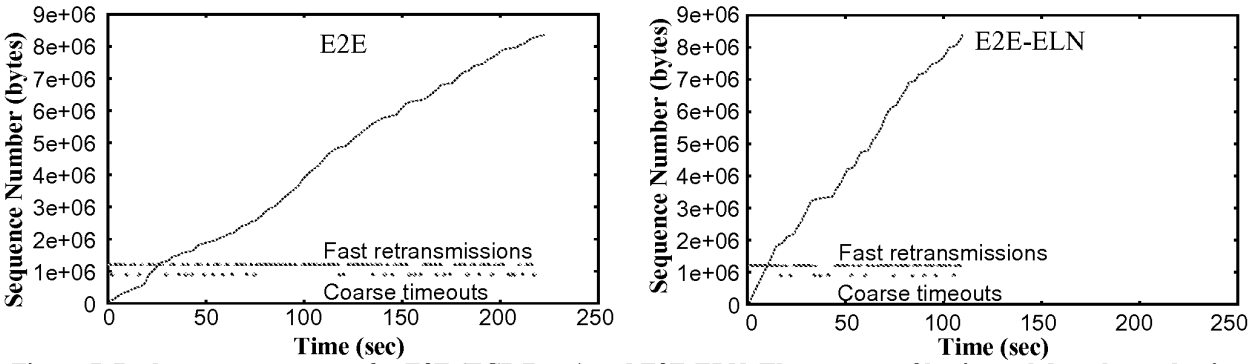


Figure 7. Packet sequence traces for E2E (TCP Reno) and E2E-ELN. The top row of horizontal dots shows the times when fast retransmissions occur; the bottom row shows the coarse timeouts.

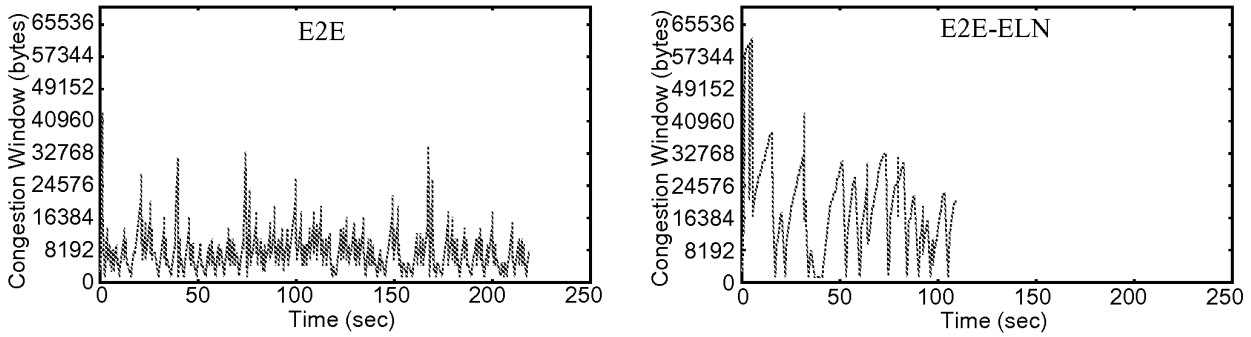


Figure 8. Congestion window size as a function of time for E2E (TCP Reno) and E2E-ELN. This figure clearly shows the utility of ELN in preventing rapid fluctuations, thereby maintaining a larger average congestion window size.

the relative performance. This is because in situations that E2E suffers a coarse timeout for a loss, the probability that E2E-NEWRENO does not, increases with the number of outstanding packets in the network.

Explicit Loss Notification: One way of eliminating the long delays caused by coarse timeouts is to maintain as large a window size as possible. E2E-NEWRENO remains in fast recovery if the new acknowledgment is only partial, but reduces the window size to half its original value upon the arrival of the first new acknowledgment. The E2E-ELN and E2E-ELN-RXMT protocols use ELN information (Section 3.1) to prevent the sender from reducing the size of the congestion window in response to a wireless loss. Both these schemes perform better than E2E-NEWRENO, and over two times better than E2E. This is a result of the sender's explicit awareness of the wireless link, which reduces the number of coarse timeouts (Figure 7) and rapid window size fluctuations (Figure 8). The E2E-ELN-RXMT protocol performs only slightly better than E2E-ELN when the socket buffer size is 32 KB. This is because there is usually enough data in the pipe to trigger a fast retransmission for E2E-ELN. The performance benefits of E2E-ELN-RXMT are more pronounced when the socket buffer size is smaller, as the numbers for the 8 KB socket buffer size indicate (Table 3). This is because E2E-ELN-RXMT does not wait for three duplicate acknowledgments before retransmitting a packet, if it has ELN information for it. The maximum socket buffer size of 8 KB limits the number of unacknowledged packets to a small number at any point in time, which reduces the probability of three duplicate acknowledgments arriving after a loss and triggering a fast retransmission.

Despite explicit awareness of wireless losses, timeouts sometimes occur in the ELN-based protocols. This is a result of our implementation of the ELN protocol, which does not convey information about multiple wireless-related losses to the sender. Since it is coupled with only cumulative acknowledgments, the sender is unaware of the occurrence of multiple wireless-related losses in a window; we plan to couple SACKs and ELN together in future work. Section 5.2 discusses some possible implementation strategies and policies for ELN.

Selective acknowledgments: We experimented with two different SACK schemes. In the LAN case, we used a simple SACK scheme based on a subset of the SMART proposal. This protocol was the best of the end-to-end protocols in this situation, achieving a throughput of 1.25 Mbps (in contrast, the best local scheme, LL-SMART-TCP-AWARE, obtained a throughput of 1.39 Mbps).

In the WAN case, we based our SACK implementation [4] on RFC 2018. For the exponentially-distributed loss pattern

RED HAT 1012
RED HAT V. CAS

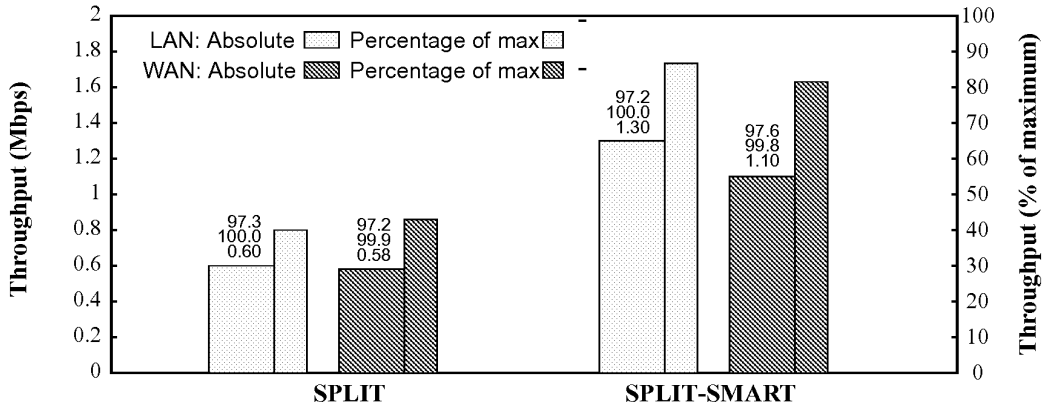


Figure 9. Performance of split-connection protocols: bit error rate = 1.9×10^{-6} (1 error/65536 bytes).

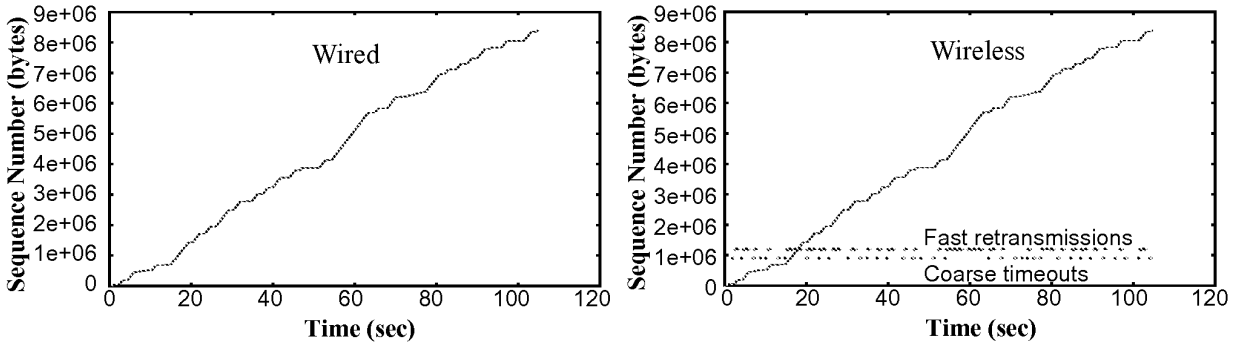


Figure 10. Packet sequence trace for the wired and wireless parts of the SPLIT protocol. The wireless part has two rows of horizontal dots: the top one shows the times of fast retransmissions and the bottom one the times of the time-out-based ones.

we used, the throughput was about 0.8 Mbps, significantly higher than the 0.31 Mbps throughput of TCP Reno. However, this is still about 35% worse than LL-OPT. Even though SACKs allow the sender to often recover from multiple losses without timing out, the sender's congestion window decreases every time there is a packet dropped on the wireless link, causing it to remain small.

In summary, E2E-NEWRENO is better than E2E, especially for large socket buffer sizes. Adding ELN to TCP improves throughput significantly by successfully preventing unnecessary fluctuations in the transmission window. Finally, SACKs provide significant improvement over TCP Reno, but perform about 10-15% worse than the best link-layer schemes in the LAN experiments, and about 35% worse in the WAN experiments. These results suggest that an end-to-end protocol that has both ELN and SACKs will result in good performance, and is an area of current work.

4.4 Split-Connection Protocols

The main advantage of the split-connection approaches is that they isolate the TCP source from wireless losses. The TCP sender of the second, wireless connection performs all the retransmissions in response to wireless losses.

	SPLIT	SPLIT-SMART
LAN (8 KB)	0.54 (97.4%,100%)	1.30 (97.6%,100%)
LAN (32 KB)	0.60 (97.3%,100%)	1.30 (97.2%,100%)
WAN (32 KB)	0.58 (97.2%,100%)	1.10 (97.6%,100%)

Table 4. Summary of results for the split-connection schemes at an average error rate of 1 every 64 KB.

Figure 9 and Table 4 show the throughput and goodput for the split connection approach in the LAN and WAN environments. We report the results for two cases: when the wireless connection uses TCP Reno (labeled SPLIT) and when it uses the SMART-based selective acknowledgment scheme described earlier (labeled SPLIT-SMART). We see that the throughput achieved by the SPLIT approach (0.6 Mbps) is quite low, about the same as that for end-to-end TCP Reno (labeled E2E in Figure 6). The reason for this is apparent from Figures 10 and 11, which show the progress of the data transfer and the size of the congestion window for the wired and wireless connections. We see that the wired connection neither has any retransmissions nor any timeouts, resulting in a wired goodput of 100%. However, it (eventually) stalls whenever the sender of the wireless connection experiences a timeout, since the amount of buffer

RED HAT 1012
RED HAT V. CAS

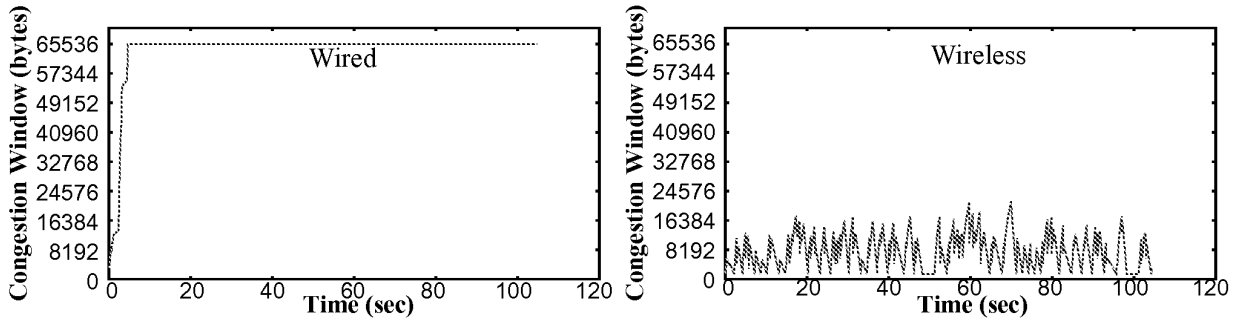


Figure 11. Congestion window sizes as a function of time for the wired and wireless parts of the split TCP connection. The wired sender never sees any losses and maintains a 64 KB congestion window. However, the wireless TCP connection's congestion window fluctuates rapidly.

space at the base station (64 KB in our experiments) is bounded[3]. In the WAN case, the throughput of the SPLIT approach is about 0.58 Mbps which is better than the 0.31 Mbps that the E2E approach achieves (Figure 6), but not as good as several other protocols described earlier. The large congestion window size of the wired sender in SPLIT enables a higher bandwidth utilization over the wired network, compared to an end-to-end TCP connection where the congestion window size fluctuates rapidly.

As expected, the throughput for the SPLIT-SMART scheme is much higher. It is about 1.3 Mbps in the LAN case and about 1.1 Mbps in the WAN case. The SMART-based selective acknowledgment scheme operating over the wireless link performs very well, especially since no reordering of packets occurs over this hop. However, there are a few times when both the original transmission and the first retransmission of a packet get lost, which sometimes results in a coarse timeout (as described in Section 3.1). This explains the difference in throughput between the SPLIT-SMART scheme and the LL-SMART-TCP-AWARE scheme (Figure 3).

In summary, while the split-connection approach results in good throughput if the wireless connection uses special mechanisms, the performance is worse than that of a well-tuned, TCP-aware link-layer protocol (LL-TCP-AWARE or LL-SMART-TCP-AWARE). Moreover, the link-layer protocol preserves the end-to-end semantics of TCP acknowledgments. This demonstrates that the end-to-end connection need not be split at the base station in order to achieve good performance.

3. A larger buffer at the base station will not necessarily improve performance for two reasons: (1) we measure performance in terms of receiver throughput, which is limited by the small congestion window size of the wireless connection, and (2) a long enough transfer will still fill up the buffer.

Burst Length	LL-TCP-AWARE (Mbps)	LL-SMART-TCP-AWARE (Mbps)
2	1.25	1.28
4	1.02	1.20
6	0.84	1.10

Table 5. Throughputs of LL-TCP-AWARE and LL-SMART-TCP-AWARE at different burst lengths. This illustrates the benefits of SACKs, even for a high-performance, TCP-aware link protocol.

4.5 Reaction to Burst Errors

In this section, we report the results of some experiments that illustrate the benefit of selective acknowledgments in handling burst losses. We consider two of the best performing local protocols: LL-TCP-AWARE (Snoop) and LL-SMART-TCP-AWARE (Snoop with SMART-based selective acknowledgments). LL-TCP-AWARE recovers from a single loss by retransmitting the lost packet when two duplicate acknowledgments arrive for it. It also keeps track of the number of expected duplicate acknowledgments and the next expected new acknowledgment after this local retransmission. If this loss is part of a burst, the first new acknowledgment to arrive after the duplicates will be less than the next expected new one; this causes an immediate retransmission of the lost segment. This is similar to the mechanism used by E2E-NEWRENO (Section 3.1). LL-SMART-TCP-AWARE uses the additional useful information provided by the SMART scheme — the sequence number of the segment that caused the duplicate acknowledgment — to accurately determine losses and recover from them.

Table 5 shows the performance of the two protocols for bursts of lengths 2, 4, and 6 packets. These errors are generated at an average rate of one every 64 KBytes of data, and 2, 4, or 6 packets are destroyed in each case. Selective acknowledgments improve the performance of LL-SMART-TCP-AWARE over LL-TCP-AWARE by up to 30% in the presence of burst errors. While this is a fairly

RED HAT 1012
RED HAT V. CAS

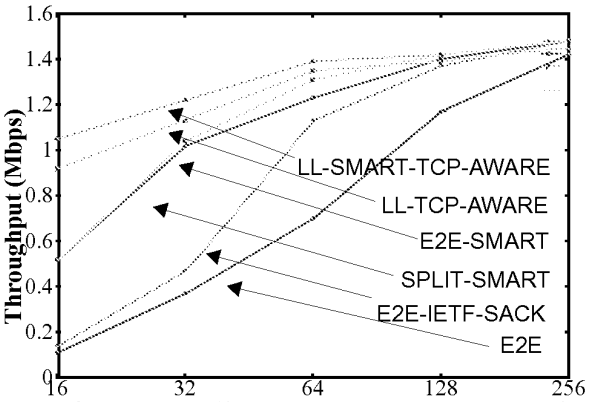


Figure 12. Performance of six protocols (LAN case) across a range of bit-error rates, ranging from 1 error every 16 KB to 1 every 256 KB shown on a log-scale.

simplistic burst-error model, it does illustrate the problems caused by the loss of multiple packets in succession. We are in the process of experimenting with a *temporal* burst-loss model based on average lengths of fades and other causes of wireless losses. The parameters of this model are derived from a trace-based modeling and characterization of the WaveLAN network [23].

4.6 Performance at Different Error Rates

In this section, we present the results of several experiments performed across a range of bit-error rates, for some of the protocols described earlier — E2E (the baseline case), LL-TCP-AWARE, LL-SMART-TCP-AWARE, E2E-SMART, E2E-IETF-SACK, and SPLIT-SMART. We chose the best performing protocols from each category, as well as some other protocols (e.g., E2E-IETF-SACK) to illustrate some interesting effects.

Figure 12 shows the performance of these protocols for an 8 MByte end-to-end transfer in a LAN environment, across exponentially distributed error rates ranging from 1 error every 16 KB to 1 error every 256 KB, in increasing powers of two. We find that the overall qualitative results and conclusions are similar to those presented earlier for the 64 KB error rate. At low error rates (128 KB and 256 KB points in the graph), all the protocols shown perform almost equally well in improving TCP performance. At the 16 KB error rate, the performance of the TCP-aware link-layer schemes is about 1.75-2 times better than E2E-SMART and about 9 times better than TCP Reno.

Another interesting point to note is the relative performance of E2E-IETF-SACK and E2E-SMART, especially at the high error rates. The congestion window does not grow larger than a few packets in the steady state at these error rates where there are multiple losses in many windows. E2E-IETF-SACK does not retransmit any packet using SACK information unless it receives three duplicate acknowledgments (to overcome potential reordering of packets in the network), which implies that no fast retransmissions are triggered if the number of packets in the window is less than four or five[4]. The sender's congestion window is often smaller than this, resulting in timeouts and degraded performance. In contrast, our implementation of E2E-SMART assumes no reordering of packets (which is justified in the LAN case) and retransmits the lost packet when the first duplicate acknowledgment with loss information arrives. This reduces the number of timeouts and results in better end-to-end performance. In Section 5.3, we outline a scheme in which the IETF protocol can be modified to work well even when the sender's congestion window is not large enough to provide enough duplicate acknowledgments.

5. Discussion

In this section, we present a discussion of some miscellaneous issues. We discuss the effects of handoff on TCP performance, some implementation strategies and policies for the ELN mechanism introduced in Section 3.1, and some issues related to SMART-based and IETF selective acknowledgment schemes.

5.1 Wireless Handoffs

Wireless networks are usually organized in a cellular topology where each cell includes a base station that acts as a router between the wireless subnet and a wireline backbone. Mobile hosts typically communicate via the base station in the cell they are currently located in. Examples of networks organized in this fashion include cellular telephone networks and wireless local-area networks.

As a mobile host moves, it may get out of the range of its current base station but still be within the range of other neighboring base stations. To maintain the mobile host's connectivity, a *handoff* procedure is invoked to re-route traffic to and from the mobile host via the new base station. However, depending on the details of the handoff algorithms, this procedure could lead to packet losses and reordering, which in turn could cause significant deterioration in the performance of ongoing TCP transfers [8].

Several proposals have been made for achieving fast handoffs. Two examples include multicast-based handoffs [25] and hierarchical handoffs [9]. In both these schemes, handoffs are made fast by restricting updates to the immediate vicinity of the mobile host. As a result the handoff latency in a WaveLAN-based wireless local-area network is of the order of 10-30 ms.

4. This depends on whether delayed acknowledgments are used.

RED HAT 1012
RED HAT V. CAS

A small amount of buffering and retransmission from base stations prevents packet loss during the short handoff period. In [9], the buffering happens at the mobile host's old base station, which forwards packets to the new base station at the time of handoff. In [25], one or more base stations in the vicinity join a multicast group corresponding to the mobile host and receive all packets destined to it, in anticipation of a handoff. When the handoff happens, the new base station is readily able to forward the buffered and the newly arriving packets without introducing any reordering, thereby preventing unnecessary invocations of TCP fast retransmissions. Experimental results reported in [25] indicate that such fast handoffs have a minimal adverse effect on TCP performance, even when the handoff frequency is as high as once per second.

In contrast to the above schemes that operate at the network layer, handoffs in a split-connection context, such as in I-TCP [3], involve the transfer of transport-layer state from the old base station to the new one. This results in significantly higher latency; for example, [2] reports I-TCP handoff latencies of the order of hundreds of milliseconds in a WaveLAN-based network.

5.2 Implementation Strategies for ELN

Section 3.1 described the ELN mechanism by which the transport protocol can be made aware of losses unrelated to network congestion and react appropriately to such losses. In this section, we outline possible implementation strategies and policies for this mechanism.

A simple strategy for implementing ELN would be to do so at the receiver, as we did for the results presented in this paper. In this method, the corruption of a packet at the link-layer, indicated by a CRC error, is passed up to the transport layer, which sends an ELN message with the duplicate acknowledgments for the lost packet. In practice, it may be hard to determine the connection that a corrupted packet belongs to, since the header could itself be corrupted: this can be handled by protecting the TCP/IP header using an FEC scheme. However, there are circumstances in which entire packets, including link-level headers, are dropped over a wireless link. In such circumstances, the base station generates ELN messages to the sender (in-band, as part of the acknowledgment stream) when it observes duplicate TCP acknowledgments arriving from the mobile host.

We expect Explicit Loss Notifications to be useful in the context of multi-hop wireless networks, and are exploring this in on-going work. Such networks (e.g., Metricom's Ricochet network [21]) typically use packet radio units to route packets to and from a wired infrastructure. Here, in order to implement ELN, periodic messages are exchanged between adjacent packet radio units about queue lengths and this information is used as a heuristic to distinguish between congestion and packet corruption, especially when entire packets (including headers) are corrupted or dropped over a wireless link. This, coupled with a simple link-level scheme to convey NACK information about missing packets, is sufficient to generate ELN messages to the source.

5.3 Selective Acknowledgment Issues

Our experience with the IETF SACK scheme highlights some weaknesses with it both when sender window sizes are small. This situation can be improved by enhancing the sender's loss recovery algorithm as follows. In general, the arrival of one duplicate acknowledgment at the receiver indicates that one segment has successfully reached the receiver. Rather than wait for three duplicate acknowledgments and perform a fast retransmission, the sender now transmits a *new* segment from beyond the "right edge" of the current window upon the arrival of the first and second duplicate acks. This probes the network for sustained congestion and generates duplicate acknowledgments. Note that we have not violated standard congestion control procedures by doing this: we only send out a segment when one has left the data pipe, following the principle of conservation of packets [13]. This enhancement can coexist with SACKs to further avoid timeouts, since the arrival of an acknowledgment with a SACK block indicating the reception of the newly transmitted segment is a strong indicator that the original segment was lost, independent of whether three duplicate acknowledgments arrive or not. Thus, this mechanism will improve performance when the sender's window is small and losses occur, and is further explored and described in [6].

6. Conclusions

In this paper, we have presented a comparative analysis of several techniques to improve the end-to-end performance of TCP over lossy, wireless hops. We categorize these techniques as end-to-end, link-layer or split-connection based. We use the end-to-end throughput, and the wired and wireless goodputs as metrics for comparison.

Our results lead to the following conclusions:

1. A reliable link-layer protocol that uses knowledge of TCP (LL-TCP-AWARE) to shield the sender from duplicate acknowledgments arising from wireless losses gives a 10-30% higher throughput than one (LL) that operates independently of TCP and does not attempt in-order delivery of packets. Also, the former avoids redundant retransmissions by both the sender and the base station, resulting in a higher goodput. Of the schemes we investigated, the TCP-aware link-layer protocol with selective acknowledgements performs the best.

2. The split-connection approach, with standard TCP used for the wireless hop, shields the sender from wireless losses. However, the sender often stalls due to timeouts on the

RED HAT 1012
RED HAT V. CAS

wireless connection, resulting in poor end-to-end throughput. Using a SMART-based selective acknowledgment mechanism for the wireless hop yields good throughput. However, the throughput is still slightly less than that for a well-tuned link-layer scheme that does not split the connection. This demonstrates that splitting the end-to-end connection is not a requirement for good performance.

3. The SMART-based selective acknowledgment scheme we used is quite effective in dealing with a high packet loss rate when employed over the wireless hop or by a sender in a LAN environment. In the WAN experiments, the SACK scheme based on the IETF Draft resulted in significantly improving end-to-end performance, although its performance was not as good as in the best link schemes. From our results we conclude that selective acknowledgment schemes are very useful in the presence of lossy links, especially when losses occur in bursts.

4. End-to-end schemes, while not as effective as local techniques in handling wireless losses, are promising since significant performance gains can be achieved without any extensive support from intermediate nodes in the network. The explicit loss notification scheme we evaluated resulted in a throughput improvement of more than a factor of two over TCP-Reno, with comparable goodput values.

7. Future Work

Our experiments with various SACK and ELN mechanisms demonstrate the significant benefits of such schemes, as described in Section 5. We are in the process of evaluating protocol enhancements based on these ideas in the presence of both network congestion and wireless losses in different network topologies, especially in networks with multiple wireless hops. In addition, we are evaluating the performance of several of the protocols described in this paper under other patterns of loss derived from traces in [23].

We are investigating the impact of large variations in connection round-trip times and the impact of bandwidth and latency asymmetry on transport performance [5]. Large round-trip variations are common in networks like the Metricom Ricochet wireless network [21], especially in the presence of bidirectional traffic. Bandwidth asymmetry is prevalent in many cable and satellite networks with low-bandwidth return channels.

8. Acknowledgments

We are grateful to Steven McCanne and the anonymous reviewers for ACM SIGCOMM '96 and IEEE/ACM Transactions on Networking for several comments and suggestions that helped improve the quality of this paper. We thank Sally Floyd and Vern Paxson for useful discussions on SACKs and related topics.

This work was supported by DARPA contract DAAB07-95-C-D154, by the State of California under the MICRO program, and by the Hughes Aircraft Corporation, Metricom, Fuji Xerox, Daimler-Benz, Hybrid Networks, and IBM. Hari is partially supported by a research grant from the Okawa Foundation.

9. References

[1] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani, and R. D. Gitlin. AIRMAIL: A Link-Layer Protocol for Wireless Networks. *ACM ACM/Baltzer Wireless Networks Journal*, 1:47–60, February 1995.

[2] A. Bakre and B. R. Badrinath. Handoff and System Support for Indirect TCP/IP. In *Proc. Second Usenix Symp. on Mobile and Location-Independent Computing*, April 1995.

[3] A. Bakre and B. R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In *Proc. 15th International Conf. on Distributed Computing Systems (ICDCS)*, May 1995.

[4] H. Balakrishnan. An Implementation of TCP Selective Acknowledgments. ftp://daedalus.cs.berkeley.edu/pub/tcpsack/, 1996.

[5] H. Balakrishnan, V. N. Padmanabhan, and R.H. Katz. The Effects of Asymmetry on TCP Performance. In *Proc. ACM MOBICOM '97*, September 1997.

[6] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, M. Stemm, and R.H. Katz. TCP Behavior of a Busy Web Server: Analysis and Improvements. Technical Report UCB/CSD-97-966, University of California at Berkeley, 1997.

[7] H. Balakrishnan, S. Seshan, and R.H. Katz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks*, 1(4), December 1995.

[8] R. Caceres and L. Iftode. Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments. *IEEE Journal on Selected Areas in Communications*, 13(5), June 1995.

[9] R. Caceres and V. N. Padmanabhan. Fast and Scalable Handoffs in Wireless Internetworks. In *Proc. 1st ACM Conf. on Mobile Computing and Networking*, November 1996.

[10] A. DeSimone, M. C. Chuah, and O. C. Yue. Throughput Performance of Transport-Layer Protocols over Wireless LANs. In *Proc. Globecom '93*, December 1993.

RED HAT 1012
RED HAT V. CAS

[11] K. Fall and S. Floyd. Simulation-based Comparisons of Tahoe, Reno, and Sack TCP. *Computer Communications Review*, 1996.

[12] J. C. Hoe. Start-up Dynamics of TCP's Congestion Control and Avoidance Schemes. Master's thesis, Massachusetts Institute of Technology, 1995.

[13] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM 88*, August 1988.

[14] V. Jacobson and R. T. Braden. *TCP Extensions for Long Delay Paths*. RFC, Oct 1988. RFC-1072.

[15] P. Karn. The Qualcomm CDMA Digital Cellular System. In *Proc. 1993 USENIX Symp. on Mobile and Location-Independent Computing*, pages 35–40, August 1993.

[16] P. Karn and C. Partridge. Improving Round-Trip Time Estimates in Reliable Transport Protocols. *ACM Transactions on Computer Systems*, 9(4):364–373, November 1991.

[17] S. Keshav and S. Morgan. SMART Retransmission: Performance with Overload and Random Losses. In *Proc. Infocom '97*, 1997.

[18] S. Lin and D. J. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Inc., 1983.

[19] Mathis, M. and Mahdavi, J. and Floyd, S. and Romanow, A. *TCP Selective Acknowledgment Options*, 1996. RFC-2018.

[20] S. McCanne and V. Jacobson. The BSD Packet Filter: A New Architecture for User-Level Packet Capture. In *Proc. Winter '93 USENIX Conference*, San Diego, CA, January 1993.

[21] Metricom, Inc. http://www.metricom.com, 1997.

[22] S. Nanda, R. Ejzak, and B. T. Doshi. A Retransmission Scheme for Circuit-Mode Data on Wireless Links. *IEEE Journal on Selected Areas in Communications*, 12(8), October 1994.

[23] G. T. Nguyen, R. H. Katz, B. D. Noble, and M. Satyanarayanan. A Trace-based Approach for Modeling Wireless Channel Behavior. In *Proc. Winter Simulation Conference*, Dec 1996.

[24] J. B. Postel. *Transmission Control Protocol*. RFC, Information Sciences Institute, Marina del Rey, CA, September 1981. RFC-793.

[25] S. Seshan, H. Balakrishnan, and R. H. Katz. Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience. *Kluwer Journal on Wireless Personal Communications*, January 1997.

[26] W. R. Stevens. *TCP/IP Illustrated, Volume 1*. Addison-Wesley, Reading, MA, Nov 1994.

[27] AT&T WaveLAN: PC/AT Card Installation and Operation. AT&T manual, 1994.

[28] R. Yavatkar and N. Bhagwat. Improving End-to-End Performance of TCP over Mobile Internetworks. In *Mobile 94 Workshop on Mobile Computing Systems and Applications*, December 1994.

Hari Balakrishnan (S '95 / ACM S '95) is a Ph.D. candidate in Computer Science at the University of California at Berkeley. His research interests are in the areas of computer networks, wireless and mobile computing, and distributed computing and communication systems. His current research is in the area of reliable data transport over heterogeneous networking technologies.

Hari received a B. Tech. degree in Computer Science and Engineering from the Indian Institute of Technology, Madras, in 1993 and an M.S. degree in Computer Science from Berkeley in 1995. He received best student paper awards at the Winter Usenix '95 and at the ACM Mobicom '95 conferences, and is the recipient of a research grant from the Okawa Foundation. On the WWW, his URL is http://www.cs.berkeley.edu/~hari and his e-mail address is hari@cs.berkeley.edu.

Venkata N. Padmanabhan (IEEE S '94 / ACM S '94) is a Ph.D. candidate in Computer Science at the University of California at Berkeley. He received his B.Tech. degree from the Indian Institute of Technology, Delhi in 1993 and his M.S. degree from the University of California at Berkeley in 1995, both in Computer Science.

Venkat has done research in the areas of Computer Networking, Mobile Computing and Operating Systems. The focus of his current work is network support for efficient Web access, and data transport over asymmetric networks. He received the best student paper award at the Usenix '95 conference. He may be reached via e-mail at padmanab@cs.berkeley.edu and on the Web at http://www.cs.berkeley.edu/~padmanab.

Srinivasan Seshan (ACM M '92) received a B.S. in Electrical Engineering, an M.S., and a Ph.D. in Computer Science from the University of California at Berkeley in 1990, 1993 and 1995 respectively. Since 1995, he has been a research staff member at the IBM T.J. Watson Research Center. His research interests include computer networks, mobile computing and distributed computing. His e-mail address is srini@watson.ibm.com and his WWW home page is at http://www.research.ibm.com/people/s/srini.

Randy H. Katz (F '96, ACM F '96) is a professor of computer science at the University of California at Berkeley, and is a principal investigator in the Bay Area Research Wireless Access Network (BARWAN) project. He has taught at Berkeley since 1983, with the exception of 1993 and 1994 when he was a program manager and deputy director of the Computing Systems Technology Office at the Defense Department's Advanced Research Projects Agency. He has written over 130 technical publications on CAD, database

RED HAT 1012
RED HAT V. CAS

management, multiprocessor architectures, high performance storage systems, video server architectures, and computer networks.

Dr. Katz received a B.S. degree at Cornell University, and an M.S. and a Ph.D. at the University of California at Berkeley, all in computer science. His e-mail address is randy@cs.berkeley.edu and his WWW home page is http://www.cs.berkeley.edu/~randy.

RED HAT 1012
RED HAT V. CAS

An Efficient Multipath Forwarding Method

Johnny Chen Peter Druschel Devika Subramanian
Department of Computer Science
Rice University
Houston, TX 77005
Email: {jc,druschel,devika}@cs.rice.edu
Phone: (713) 527-4664, Fax: (713) 285-5930

Abstract— **We motivate and formally define dynamic multipath routing and present the problem of packet forwarding in the multipath routing context. We demonstrate that for multipath sets that are *suffix matched*, forwarding can be efficiently implemented with (1) a per packet overhead of a small, fixed-length path identifier, and (2) router space overhead linear in K, the number of alternate paths between a source and a destination. We derive multipath forwarding schemes for suffix matched path sets computed by both de-centralized (link-state) and distributed (distance-vector) routing algorithms. We also prove that (1) distributed multipath routing algorithms compute suffix matched multipath sets, and (2) for the criterion of ranked k-shortest paths, de-centralized routing algorithms also yield suffix matched multipath sets.**

Keywords— **Routing, Multipath, Forwarding.**

I. Introduction

Multipath routing aims to exploit the connectivity of the underlying physical networks by providing multiple paths between source-destination pairs. Unlike in single shortest path routing, the originating node has a choice at any given time of potentially more than one path to a particular destination. In a connectionless network, we can have per-packet multipath routing where an originating node sends individual packets to a given destination on different paths. In a connection-oriented network, we can have per-connection multipath routing where an originating node uses different paths for different connections to the same destination and packets on each connection travel the same path.

Multipath routing has the potential to aggregate bandwidth, allowing a network to support data transfer rates higher than what is possible with any one path [22]. In connection-oriented networks with QoS guarantees, they reduce blocking probabilities [12]. Multipath routing algorithms are able to multiplex connectionless traffic, thereby proactively reducing routing oscillation and alleviating congestion [3, 10]. Finally, the use of independent paths can increase the reliability of packet delivery [12].

There are two challenges in designing adaptive multipath routing algorithms. The first is devising efficient distributed or de-centralized algorithms for computing good multipath sets between pairs of nodes in the network. The second challenge is the forwarding of packets. Since each node has more than one path to a given destination, packet forwarding procedures at each node need to correctly and efficiently forward packets along their intended paths. Our chief technical contribution in this paper is an efficient solution to multipath forwarding. We demonstrate that for an interesting class of multipath sets, called *suffix matched* path sets, we can use a fixed length path identifier per packet and have router space requirements proportional to the number of paths calculated.

The traditional use of multipath routing (e.g., in the OSPF protocol) has been limited to the exploitation of a very special class of multipath sets, i.e., paths of equal cost. The multipath routing algorithms we consider in this paper allow more general multipath sets and are computed dynamically to adapt to changes in the network. As an example, consider a multipath routing method that maintains at each node the three best paths (not necessarily of equal cost) between that node and each available destination in the network. The originating node in such a network can decide whether to send a packet or a set up a virtual connection to a particular destination along its 1st, 2nd or the 3rd best path.

The paper is organized as follows. In the next section, we formally define adaptive multipath routing and the associated data packet forwarding problem. We then present the *suffix matched* class of multipath sets, a class for which the forwarding problem can be solved efficiently. In Section II, we describe the basis of our new forwarding method for suffix matched path sets and show that it uses (1) a per packet overhead of a small, fixed-length path identifier, and (2) router space overhead linear in K, the number of calculated paths between source/destination pairs. In

Appears in Proc. of IEEE Infocom'98, San Francisco, CA, March 1998.

RED HAT 1012
RED HAT V. CAS

Sections III and IV, we derive multipath forwarding schemes for suffix matched path sets computed by the distance vector and link state algorithms. We also prove that (1) distance-vector algorithm compute suffix matched multipath sets, and (2) for the criterion of ranked k-shortest paths, the link state algorithm yields suffix matched multipath sets. We conclude with related work in forwarding and multipath routing and discuss current work on handling non-suffix matched path sets.

II. Multipath Routing and Forwarding

We model a network as a graph $G = (V, E)$, where V is the set of nodes in the network and E is the set of links or edges. Each link $(x_i, x_j) \in E$ has an associated positive cost $c_{x_i x_j}$. A path p in G is a list of nodes $(x_1, \ldots, x_n)$ such that $\forall i, 1 \leq i < n$, $(x_i, x_{i+1}) \in E$. Path p has cost $\sum_{i=1}^{n-1} c_{x_i x_{i+1}}$. No node appears more than once in a *simple* or loop-free path. To route packets between nodes in a network, each node $x \in V$ implements two functions:

$$h_x : V \rightarrow \Lambda$$

$$\mathcal{G}_x : \Lambda \rightarrow V$$

The *path selection* function h_x chooses a path identifier ψ (on a per-packet or per-connection basis) for a given destination node $d \in V$ from the set Λ. Path identifiers encode distinct paths from x to d; they are names for paths in the network. We use path identifiers to name paths so that we can use a fixed name ψ for a path, while having its composition change with time. The forwarding function $\mathcal{G}_x$ takes a path identifier $\psi \in \Lambda$ and provides the next hop, i.e., the node following x in the path that is encoded by the identifier ψ. As an example, if node x wishes to send a packet to node d on the path named ψ, it adds the label ψ to the packet, and forwards it to $\mathcal{G}_x(\psi)$: the next hop on the path named ψ from x to d.

Single shortest path routing is a special case where there is no need to have a path identifier set Λ. In this path setting, the path selection function is the identity function: $h_x(d) = d$, $d \in V$. This is because exactly one path p from x to d is used at any given time — there is no choice of paths.[1] By the Bellman optimality principle, the shortest path from any intermediate node n in p to node d is the subpath of p from n to d; thus we can uniquely identify the path p along any node in p by the destination address d. We can replace the path identifier argument by the destination and simplify the forwarding function in a single shortest path setting to be $\mathcal{G}_x : V \rightarrow V$ which identifies the next hop for a given destination. $\mathcal{G}_x$ is implemented by forwarding tables at node x. The selection function h_x is normally evaluated once for a given packet at the originating host/router.

In a multipath setting, an originating node x has a choice of several paths to a particular destination at any given time (i.e, the range of the function h_x is a set of size ≥ 1). The particular path $h_x(d)$ chosen by x for a destination d is a function of the specific objectives of the routing system as a whole. We will have little to say about these functions in this paper. Examples of objectives include: increasing end-to-end path capacity (bandwidth) [15], providing QoS guarantees [22], or reducing service blocking probabilities [12].

The forwarding functions $\mathcal{G}_x$, for each $x \in V$ need to respect the following constraint:

$\forall \psi \in \Lambda$ such that ψ identifies path $= (x_1, \ldots x_n)$, $\mathcal{G}_{x_i}(\psi) = x_{i+1}, 1 \leq i < n$.

Efficiently enforcing the above constraint on the $\mathcal{G}_x$ functions forms the crux of the multipath forwarding problem.

To implement the $\mathcal{G}$ functions, we need a scheme for constructing path identifiers that unambiguously identify paths between nodes in a network. One common approach, called source routing, is to use the path description itself as its identifier. In this method, path identifiers are of variable length and the overhead of stamping individual packets with path identifiers increases as the size of the network grows and the lengths of paths between nodes increase. The implementation of the $\mathcal{G}$ functions at each node x_i, for $1 \leq i < n$ requires reading the path identifier on each packet and computing $\mathcal{G}_{x_i}(x_1, \ldots, x_n)$ to be x_{i+1}. No state information (e.g., forwarding table) is needed at intermediate nodes and complete path information is needed at the sending nodes.

In this paper we focus on encoding paths with short, fixed length identifiers. As an example, if K best paths are maintained between each source and destination pair in an N node network, a path can be uniquely identified by the triple (s, d, i), where $s, d, \in V$ and $1 \leq i \leq K$. The space required for all the forwarding tables in the network is $O(KN^2 * N) = O(KN^3)$ which is the size of the mapping from each path identifier to the next hop (there are KN^2 paths), and this is maintained by each node (there are N nodes) in the network. The space require-

[1] The actual identity of the shortest path between two nodes in the network may change as the network changes, but at any given time a node can route to a destination along exactly one path.

RED HAT 1012
RED HAT V. CAS

ment of this method can be reduced by observing that a node only needs to maintain paths that pass through it. Let L be the average path length, then each path passes through L nodes on average. Therefore it suffices that each node maintains only KNL path identifiers on average, reducing the total space requirements on forwarding tables to $O(KN^2L)$. In contrast, single shortest path routing imposes a fixed overhead on a per packet basis (each packet is labeled with the destination address), and the forwarding tables in the network need $O(N^2)$ space.

Is it possible to forward packets efficiently in a multipath setting with packets annotated by short fixed-length path identifiers and with space overhead for forwarding tables no more than $O(KN^2)$, which is K times the cost for forwarding in single shortest path systems? We answer this question in the affirmative for a very interesting class of multipath sets called *suffix matched* path sets.

Suffix matched path sets: A path set P is *suffix matched* iff for all paths $(x_1, \ldots, x_n) \in P$, $\forall i, 1 \leq i < n$, $(x_i, \ldots, x_n)$ is also in P.

Consider the path set P consisting of the single shortest paths between every pair of nodes in a network. The Bellman optimality principle ensures that P is suffix matched. If the shortest path p from node x_1 to x_n is $(x_1, \ldots, x_n)$, then for $1 < i \leq n$, the shortest path from x_i to x_n is the subpath $(x_i, \ldots, x_n)$ of p.

Proposition: Packets in an N node network with a suffix matched multipath set P can be forwarded correctly with forwarding table space $O(KN^2)$ where no more than K paths are maintained between any pair of nodes.

Proof: Let the path identifier for path $p = (x_1, \ldots, x_n)$ from x_1 to x_n in a suffix matched multipath set P be $f_{x_1}(x_n)$. Let the path identifier for the suffix paths $p_i = (x_i, \ldots, x_n)$ for $1 < i \leq n$ be $f_{x_i}(x_n)$. Recall that all suffixes of p are in P by the definition of suffix matched multipath sets.

To implement path forwarding correctly, we need for node x_i to know that the suffix of its path $f_{x_i}(x_n)$ (the path $(x_{i+1}, \ldots, x_n)$) at its neighbor x_{i+1} is called $f_{x_{i+1}}(x_n)$. Therefore we need an identifier translation or swapping function $\mathcal{F}_{x_i} : \Lambda \rightarrow \Lambda$ at each node that maps the path identifier $f_{x_i}(x_n)$ of suffix path p_i to the path identifier $f_{x_{i+1}}(x_n)$ for $1 \leq i \leq n$. That is

$$f_{x_{i+1}}(x_n) = \mathcal{F}_{x_i}(f_{x_i}(x_n)) \text{ and}$$

$$x_{i+1} = \mathcal{G}_{x_i}(f_{x_i}(x_n)).$$

In other words, when node x_i receives a packet with identifier $f_{x_i}(x_n)$ from its neighbor x_{i-1}, it forwards it to node $x_{i+1} = \mathcal{G}_{x_i}(f_{x_i}(x_n))$ along with the new identifier $f_{x+1}(x_n) = \mathcal{F}_{x_i}(f_{x_i}(x_n))$. Each x_j , $i < j < n$, performs the same two operations until the packet reaches x_n.

Notice that the definition of a suffix matched path set in a distributed environment implies that if router x_0 computes the path $(x_0, \ldots, x_n)$, then router x_i, $\forall i, 0 < i < n$, x_i must also compute the path $(x_i, \ldots, x_n)$. This ensures that there exists a $\mathcal{F}_{x_i}$ implemented by x_i that is guaranteed to translate a path identifier of x_i to x_{i+1}'s corresponding path identifier.

The space requirements for the implementation of the mapping $\mathcal{F}$ is proportional to the total number of paths in the network, i.e. $O(KN^2)$. This is because each path $O(KN^2)$ is matched with at most one other path, therefore the matching table is at most $2KN^2$. The space requirement for the mapping $\mathcal{G}$ is simply the number of paths $O(KN^2)$ □.

In the transient state when suffix matched path sets are in the process of being computed at each router, correct path forwarding is not guaranteed. This is because transient path sets are not guaranteed to be suffix matched. In fact, this absence of forwarding guarantees during route transitions effects most distributed routing algorithms, whether single or multipath.

III. Distance Vector Extension

In this section, we describe a multipath extension to a Distance Vector (DV) based routing algorithm. We first review the basic DV algorithm and then extend the algorithm to compute multiple paths. We then prove that the new multipath DV algorithm computes suffix matched paths sets. Finally, we apply our forwarding method to the extended DV algorithm, show its space complexity, and prove its correctness.

A. The Distance Vector Algorithm

Distance Vector is a distributed adaptive algorithm that computes shortest path between all pairs of nodes in a dynamic network. It is based on a centralized method known as the Bellman-Ford [4, 8] algorithm, and is the basis of many practical routing algorithms in current use [18, 2, 20].

Routers implement the forwarding function $\mathcal{G}$ via a lookup of the *forwarding table*. Each entry in the DV forwarding table contains three elements: the destination address, the next-hop neighbor on the known least-cost path to the destination, and the cost of the known least-cost path to the destination. To forward a packet to its destination D, the router sends the message to the next-hop neighbor specified in the

RED HAT 1012
RED HAT V. CAS

forwarding table entry for D. Every router on the path repeats this step. The destination address and the next-hop neighbor fields are used to evaluate $\mathcal{G}$ as part of the packet forwarding process. The cost field is used only by the DV algorithm.

The task of the distributed DV algorithm is to continually update the forwarding table in each router to ensure that packets travel on the shortest path to their destination. Routers construct their forwarding tables by exchanging path information with their neighbors in the form of *distance vector packets* (DVP). Distance vector packets are excerpts of a router's forwarding table. A DVP carries the identity of the originating router, and a list of distance vectors of the form (*dst_addr*, *cost*), taken from the originating router's forwarding table.

Let C_d^r be the least path cost to destination d recorded in r's forwarding table, let N_d^r be the next hop neighbor on the least cost path to d recorded in r's forwarding table, and let c_{rs} be the cost of the link from r to s. It is assumed that the path cost metric is additive, that is, the cost C of a path $(x_1, \ldots, x_n)$ is $C = c_{x_1x_2} + c_{x_2x_3} + \ldots + c_{x_{n-1}x_n}$. Initially, the forwarding tables in each router r are initialized as follows:

$$C_r^r = 0;\ \forall s : s \neq r, C_s^r = \infty$$

When a router r receives a DVP from a neighbor s, r updates its forwarding table as follows. For all destinations d in s's DVP,

$$if\ (C_d^s + c_{rs} < C_d^r\ or\ N_d^r = s)\ then\ C_d^r = C_d^s + c_{rs}$$

where $N_d^r = s$. Notice the cost added to C_d^s is c_{rs}, the cost from r to s. This addition reflects the correct path cost from r to d through s.

Routers exchange distance vector packets with all their neighbors periodically, or in response to significant changes in the forwarding table. It can be shown that after a bounded number of DVP exchanges following a topology or link cost change, all routers' forwarding tables will contain values reflecting the shortest paths to all destinations.

B. Methods of Calculating Multiple Paths in DV

The single shortest path DV algorithm can be extended to allow the calculation of multiple paths between each pair of nodes. For instance, to calculate the ranked k shortest paths to each destination, each router maintains in its forwarding table (and advertises in its DVPs) k entries for each destination. Upon receiving DVPs from its m neighbors, the router computes its own k best paths from the $m * k$ paths received from its neighbors.

Path selection criteria other than k ranked paths, such as link or node disjoint paths, can also be implemented in a DV environment. We show that any DV-based multipath routing algorithm that satisfies the following conditions calculates suffix matched path sets.

1. A router r may install a forwarding table entry with $N_d^r = s$ only upon receiving a corresponding distance vector for destination d from its neighbor s.
2. A router r never advertises a distance vector for destination d unless it has a corresponding forwarding table entry for d.

Proposition: The path set P calculated by a multipath DV routing algorithm that satisfies conditions 1 and 2 are suffix matched.

Proof: Assume the algorithm calculates a path $(x_0, \ldots, x_n) \in P$ that is not suffix matched. Then there must exist a router x_i, $0 < i < n$ such that $(x_{i-1}, \ldots, x_n) \in P$, but $(x_i, \ldots, x_n) \notin P$. That is, x_{i-1} has a forwarding table entry with $N_d^{x_{i-1}} = x_i$, but x_i does *not* have a forwarding table entry with $N_d^{x_i} = x_{i+1}$. Thus, either x_{i-1} has installed the forwarding table entry without receiving a corresponding distance vector from x_i, or x_i has advertised a distance vector without having a corresponding forwarding table entry. But, these possibilities are ruled out by conditions 1 and 2 □.

C. Multipath DV Extensions

In this section, we show how the proposed multipath forwarding method presented in Section II can be applied to any DV based multipath algorithms that satisfies conditions 1 and 2, and thus calculates suffix matched path sets.

The implementation of functions $\mathcal{F}$ and $\mathcal{G}$, defined in Section II, in a DV environment is defined as follows. Each router assigns an identifier ϕ for each of its paths to a given destination. Let $\phi_{r,d}^k$ be the ID router r assigns its kth path to destination d. Then, the path ID used by router r in our MPDV algorithm is $pid = [d,\ \phi_{r,d}^k]$. Adding ϕ and evaluating $\mathcal{F}$ and $\mathcal{G}$ during the packet forwarding process requires the following additions to the DV data structures.

- Forwarding table entries have two additional elements, ϕ_1 and ϕ_2. An entry in router r's forwarding table for the path $(r, s, \ldots, d)$ is now a five tuple $(d, C_{d,\phi}^r, s, \phi_1, \phi_2)$. ϕ_1 is r's identifier for the path $(r, s, \ldots, d)$ and ϕ_2 is neighbor s's identifier for the corresponding suffix path $(s, \ldots, d)$. Note that $N_d^r = s$. $C_{d,\phi}^r$ is the cost of r's ϕ path to d.
- DVP entries have one additional element, ϕ. A DVP originating from router r and advertising

RED HAT 1012
RED HAT V. CAS

r's path to destination d with identifier ϕ_1 is now a three tuple $(d, C^{r}_{d,\phi_1}, \phi_1)$.

Let D be the set of destinations and Φ^{r}_{d} be r's set of path identifiers to destination d. In its DVP, each router r advertises a distance vector $(d, C^{r}_{d,\phi_1}, \phi_1)$ corresponding to its forwarding table entry $(d, C^{r}_{d,\phi_1}, s, \phi_1, \phi_2)$, for each $d \in D$ and for each $\phi_1 \in \Phi^{r}_{d}$.

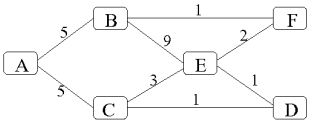
To install a route in its forwarding table upon receiving a distance vector $(d, C^{s}_{d,\phi}, \phi)$ from s, router r adds the entry $(d, C^{r}_{d,\phi_1}, s, \phi_1, \phi)$, where ϕ_1 is chosen by r to be unique in Φ^{r}_{d}, i.e., unique in its set of path identifiers to destination d. For instance, in the DV multipath algorithm for k ranked paths, r's identifier for the ith path to destination d with rank k can be chosen as k. C^{r}_{d,ϕ_1}, the cost of this path from s to d is calculated as $C^{s}_{d,\phi_1} + c_{rs}$.

The functions $\mathcal{F}$ and $\mathcal{G}$ can now be implemented directly via lookup of the forwarding table. For path ID $[d, \alpha]$, node r evaluates $\mathcal{F}_r([d, \alpha])$ as $[d, \beta]$ via lookup of the forwarding table entry for d and local identifier $\phi_1 = \alpha$, with $\beta = \phi_2$. Similarly, r evaluates $\mathcal{G}_r([d, \alpha])$ as $N^{r}_{d,\alpha}$ via lookup of the same forwarding table entry. A matching forwarding entry is guaranteed to exist when evaluating $\mathcal{F}$ and $\mathcal{G}$ because the path sets are suffix matched.

The additional storage required for the MPDV forwarding method is two path identifier fields in each forwarding table entry and one in each DVP entry. Let K be the maximal number of paths calculated to each destination, then the storage required for path identifiers is $\lceil(\log_2(K))\rceil$ bits. The additional entries in the forwarding tables and DVP for MPDV correspond to the number of alternate paths being calculated. That is, the storage overhead is a factor of $O(K)$ more than single shortest path DV. The aggregate router storage complexity of MPDV is therefore $O(KN^2)$.

D. MPDV Example

The graph in Figure 1 represents a network and the number above each link represents the cost of that link[2]. In table below are forwarding tables of selected routers to node F. The forwarding tables are computed using the Distance Vector multipath extension algorithm. For illustrative purposes, the path calculation algorithm calculates all paths with cost less than 10. Notice that this path calculation algorithm does not exclude paths that traverse a node more than once. The number of paths to any destination is at most 4, ie. $K = 4$. We assume this DV version implements split-horizon.



E's forwarding table to F

cost	Φ_1	Φ_2	next-hop
2	1	1	F
10	2	1	B
7	3	2	D
7	4	2	C

D's forwarding Table to F

cost	Φ_1	Φ_2	next-hop
3	1	1	E
6	2	2	C
8	3	4	E

C's forwarding table to F

cost	Φ_1	Φ_2	next-hop
5	1	1	E
4	2	1	D
10	3	3	E
9	4	3	D

Fig. 1. Example of MPDV.

Forwarding table construction. The process of calculating paths to F begins by F sending a DVP to its neighbors (E and B). The DVP contains the path (F, 0, 1), where F is the destination address, 0 is the cost for F to reach F, and $1 = \phi^{1}_{F,F}$. When neighbor E receives this DVP, E adds the cost $C_{EF} = 2$ to every element in F's DVP. Since the path to F has cost $2 = 0+2$, it satisfies the path calculation criteria. E keeps this path and augments its forwarding table with the entry (F, 2, 1, 1, F): F is the destination address, 2 is the cost of this path to F, the first $1 = \phi^{1}_{E,F}$ (E's name for this path to F), the second $1 = \phi^{1}_{F,F}$ (taken from F's DVP), and $F = N^{E}_{F,\phi^{1}_{F,F}}$ is this path's next-hop neighbor.

Next time E sends out its DVP, it will contain the path entry (F, 2, 1) advertising a path to F with cost 2 and $\phi^{1}_{E,F}$ of 1. When neighbor C receives this DVP containing this path to F, C augments its forwarding table with the path (F, 5, 1, 1, E) because this path is admissible under the path calculation criteria. In this example, C also chose the name of this path to be $\phi = 1$. This process of path computation continues until all routers have admissible paths to all other routers. The convergence time is proportional to the longest path between any two nodes. The complete forwarding tables to F for routers E, D, and C are shown in Figure 1.

Packet forwarding. Suppose router D wishes to send a packet to router F and D decides to send on its third best path, the one with cost 8. According to the algorithm, D tags the packet with the path ID $[F, 4]$, where F is the destination address and $4 = \phi_2$ in the forwarding entry, and forwards the packet to the next-hop neighbor E. On receiving this packet, E looks up its forwarding table and matches F and ϕ_1 with the packet's path ID. The matching entry is the last entry in E's forwarding table. Once found, E replaces the packet's path ID with [F, 2] (E's ϕ_2 for this path is 2) and forwards the packet to C. That is $\mathcal{F}_E([F, 4]) = [F, 2]$ and $\mathcal{G}_E([F, 4]) = C$. C does the same lookup and forwards the packet back to D with path ID $[F, 1]$ (here $\mathcal{F}_C([F, 2]) = [F, 1]$ and

[2]For simplicity, we assume the link costs are symmetric in this example. We note that MPDV also functions correctly with non-symmetric link costs.

RED HAT 1012
RED HAT V. CAS

$\mathcal{G}_C([F,2]) = D$). Following the algorithm, D again performs the lookup and forwards the packet to E with the path ID [F, 1]. On receiving this packet, E finally forwards the packet to F with the ID [F, 1].

For simplicity, the MPDV path calculation algorithm presented in this example did not eliminate finite looping paths. Since our path forwarding method ensures forwarding on all calculated paths, the algorithm will forward packets on finite-looping paths (eg. path (D, E, C, D, E, F)) as well as non-looping paths (eg. (D, E, F)). There are well known methods of ensuring loop-free path calculation in distance vector, refer to [14] for more detail.

IV. Link State Extension

In this section, we discuss the use of the proposed multipath forwarding method in conjunction with link-state (LS) routing algorithms. The link state algorithm, like the DV algorithm, is widely used and well understood. However, unlike DV, it relies on a centralized route calculation algorithm. A topology broadcast mechanism ensures that each router knows the current state of the entire network (i.e., topology and link cost), and routes are calculated in a centralized manner in each router. We begin with a description of the single path LS algorithm, discuss multipaths extension, and then show how our multipath forwarding method applies.

A. The Link State Algorithm

The Link State algorithm is the basis of many widely used routing algorithms. Protocols based on this algorithm are used in the Internet and in ATM networks [17, 1, 19].

In the Link State routing algorithm, every router periodically broadcasts (via flooding) its local connectivity. This information is contained in a link state packet (LSP) and consists of the router's ID, a list of its neighbor's IDs, and the cost of the connecting links. After a round of broadcasts by each router, every router has information about the entire network's topology. Each router then performs a shortest-path spanning tree computation rooted at itself. From the computation of this tree, every router knows the shortest path to all destinations in the network. This knowledge is encoded in the router's forwarding table. A LS forwarding table is a list of tuples of the form (*dst_addr*, *next-hop*), where *dst_addr* is the address of the destination and *next-hop* is the neighboring router on the shortest path to the destination.

Routers forward packets in the same way as in the DV protocol. Packets are tagged with a path ID consisting only of their destination address. Routers forward a packet by looking up their forwarding table using the packet's destination address and forwarding the packet to the specified next-hop neighbor.

B. Multipath LS Extensions

In this section, we show how the proposed forwarding method applies to link state multipath algorithms that calculate suffix matched paths. Since routers in the link state algorithm have access to the entire network topology, any centralized graph algorithm for computing multiple paths can be used. We are particularly interested in algorithms that calculate suffix matched path sets. As we will show, the k ranked paths algorithm [6], and the initial link disjoint k paths algorithm [23] are in this class. For such algorithms, our multipath forwarding algorithm can be directly applied.

As in MPDV, let $\phi^{k}_{r,d}$ be the identifier that router r assigns its kth path to destination d. The path ID used by router r is of the form $pid = [d, \phi^{k}_{d}]$. Forwarding table entries have two additional elements, ϕ_1 and ϕ_2. An entry in router r for the path $(r, s, \ldots, d)$ is a quadruple (d, s, ϕ_1, ϕ_2). ϕ_1 is r's identifier for the path $(r, s, \ldots, d)$ and ϕ_2 is neighbor s's identifier for the corresponding suffix path $(s, \ldots, d)$. Note that $N^{r}_{d,\phi_1} = s$.

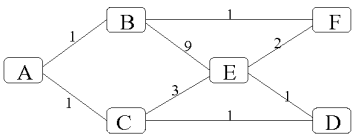
Forwarding entries are installed by the centralized link-state path calculation process. As in MPDV, ϕ_1 is chosen to be unique in r's set of paths identifiers for destination d (ie. Φ^{r}_{d}). To determine ϕ_2 router r calculates the next-hop neighbor router s's paths set for destination d. Since the path sets are assumed to be suffix matched, s's path set to d is guaranteed to include the suffix path $(s, \ldots, d)$. Then, r sets ϕ_2 equal to s's identifier ϕ_1 for that suffix path.

As in MPDV, the functions $\mathcal{F}$ and $\mathcal{G}$ can now be implemented directly via lookup of the forwarding table. Node r evaluates $\mathcal{F}_r([d, \alpha])$ as $[d, \beta]$ via lookup of the forwarding table entry for d and local identifier $\phi_1 = \alpha$, with $\beta = \phi_2$. Similarly, r evaluates $\mathcal{G}_r([d, \alpha])$ as $N^{r}_{d,\alpha}$ via lookup of the same forwarding table entry.

The number of messages exchanged by MPLS routers is exactly the same as in the traditional LS algorithm. The storage overhead per router is $O(NK)$ for the forwarding tables, where N is the number of destinations and K is the number of paths to each destination. In addition, our multipath forwarding method requires each router to calculate each of their neighbor's K paths as well. Therefore, forwarding increases the computational complexity of the multipath route calculation used in the routers by a factor m, where m is the number of neighbors.

C. Example of LS Extension

RED HAT 1012
RED HAT V. CAS



A's forwarding Table to F

Dst	next-hop	Φ_1	Φ_2
F	B	1	1
F	C	2	2
F	C	3	3

C's forwarding table to F

Dst	next-hop	Φ_1	Φ_2
F	A	1	1
F	D	2	1
F	E	3	1

E's forwarding table to F

Dst	next-hop	Φ_1	Φ_2
F	F	1	1
F	C	2	2
F	D	3	1

Fig. 2 Example of the multipath Link State Algorithm.

Consider the network shown in Figure 2. The path calculation algorithm calculated three paths from router A to F: (A, B, F), (A, C, D, E, F), (A, C, E, F), with costs 3, 5, 6 and names $\phi^1_{A,F} = 1$, $\phi^2_{A,F} = 2$, $\phi^3_{A,F} = 3$ respectively. Suppose A sends a packet to F using its third best path. Following our algorithm, A tags the packet with the ID [F,3], where F is the destination address and 3 is the ϕ_2 for that forwarding table entry. A then forwards the packet to C, the next-hop node for this path. When C receives this packet, it searches its forwarding table to find an entry with the destination address and $\phi_1 = 3$. C's third forwarding entry matches this path ID. C then replaces the packet's old path ID with [F,1] and forwards the message to E, the next-hop element of the matching entry. That is, $\mathcal{F}_C([F, 3]) = [F, 1]$ and $\mathcal{G}_C([F, 3]) = E$. Similarly, E matches the ID [F,1] and forwards the packet with the ID [F,1] to F.

D. Suffix Matched Multipath Routing Algorithms

We presented an efficient forwarding method for link state based multipath calculation algorithms that calculate suffix matching path sets. Several existing multipath algorithms generate suffix matched paths. Here we show that the k ranked path algorithm [6] produces suffix matched paths.

Proposition: The unconstrained k shortest (ranked) simple paths algorithm [6, 7] produces suffix matched path sets.

Proof: The unconstrained k shortest path algorithm uses the following invariant for computing multiple paths between all pairs of nodes in the network. A path of rank k between node x_1 and x_n is composed of subpaths all of whose ranks are $\leq k$. Since each node maintains k paths between itself and every other node, all subpaths of a k ranked path between x_1 and x_n are guaranteed to exist in the multipath set computed by the algorithm. □

The k initial link disjoint paths algorithms proposed by Topkis [23] also produces suffix matched path sets. In addition, we have already shown in Section III that any DV based multipath algorithm that satisfies two straightforward conditions produces suffix matched paths.

E. Non-Suffix Matching Paths in LS

We have shown above that several important multipath calculation algorithms produce suffix matched paths. However, many other paths selection algorithms, including algorithms that maximize flow between a pair of nodes, are not guaranteed to produce suffix matched paths. Unfortunately, our efficient multipath forwarding method cannot be directly applied to such algorithms.

We have conducted experiments with two important non-suffix matched path selection algorithms. The first algorithm computes k ranked paths subject to the constraint that the paths are link disjoint. The second algorithm calculates k ranked paths subject to capacity constraints. For this purpose, link costs reflect the capacity (i.e., bandwidth) of the link, and path costs correspond to the cost of the link on the path with least capacity. Calculations suggest that on networks with a topology typical for wide-area networks, these algorithms produce paths sets that are over 90% suffix matched. We are currently exploring extensions to our forwarding method that can take advantage of this fact and permit efficient forwarding for such multipath sets.

V. Related Work

One of the earliest multipath routing networks is the IBM SNA network [11]. The network uses global, fixed length path IDs, corresponding to different types of services. The multiple paths and path IDs require manual entries to forwarding tables. Sidhu et al. [21] present a multipath algorithm based on the distance vector algorithm. The path IDs used by this algorithm are a compressed listing of the paths, and are of variable length.

Bahk and Zarki [3] present a centralized, exponential algorithm for calculating multiple paths to reduce network load. The work does not address issue of path forwarding. Li [16] proposes a multipath routing scheme (called bifurcated routing) based on flooding. The path ID used in this algorithm is also of variable length. Currently, the only multipath routing protocol in widespread use is OSPF [5]. This algorithm uses alternate paths using only the destination address as the path ID. This implies that only multiple paths of equal or near equal cost can be used.

Optimizations to source routing have been proposed in [9, 13]. The idea is to reduce the per packet overhead by initially using source routing to establish local path identifiers, and subsequent packets only need to carry the established fixed length path

RED HAT 1012
RED HAT V. CAS

ID. Given that the path recalculation occurs infrequently, the amortized packet overhead is constant. However, this method has the memory requirement of $O(LN^2K)$, where L is the average path length, N is the number of nodes and K is the number of path calculated between nodes.

The basis of our forwarding method—namely the use of a local path identifier that is translated as a packet is forwarded through the network—has some commonalities with the way virtual circuit switching is implemented in ATM networks. However, the context in which the idea is applied and the goals are quite different. In ATM, the virtual circuit identifiers (VCI) are used to forward cells along a previously established connection (virtual circuit). Using local identifiers reduces the space requirements for the identifier in the cell header and minimizes necessary coordination among ATM switches. Our forwarding method, on the other hand, allows path forwarding without requiring additional setup. By exploiting suffix matching path sets through the use of local path identifiers, our method significantly reduces table space in the routers.

VI. Conclusion

Multipath routing has important advantages over single shortest path routing. Multipath routing is able to better exploit the underlying physical network resources, provide greater end-to-end throughput, reduce blocking probability in networks with QoS guarantees, increase reliability of packet delivery, and reduce route oscillation and congestion.

A difficult problem in multipath routing is the forwarding of packets on their intended path. Because each router individually decides where to forward packets, ensuring that packets travel a chosen path requires a form of agreement amongst all routers in the network. This agreement on paths is encoded in the form of a path ID.

In this paper, we presented a novel multipath forwarding method that uses small, fixed-length path identifiers and requires router state space proportional to the number of different paths for each destination. When compared to the general method for multipath forwarding with fixed-length identifiers, this reduces the router state space needed to enforce path agreement by a factor of L, where L is the average path length in the network. We define the *suffix matched* property of multipath sets, and show that many important multipath selection algorithms produce paths sets with this property. Our forwarding method exploits this property.

We apply our forwarding method to two well known classes of routing algorithms—the distributed distance vector algorithms, and the decentralized link state algorithms. For multipath extensions of DV, the storage and message overhead required is proportional to the number of extra paths discovered. In the LS multipath extension, no additional messages are needed. The additional computation and storage overhead for suffix matching path calculation algorithms are also proportional to the number of paths discovered.

References

[1] PNNI draft specification. ATM Forum 94-0471R13, 1994.

[2] G. Alkin and F. Baker. RIP version 2 MIB extension. RFC 1724, Xylogics, Inc., Cisco Systems, November 1994.

[3] S. Bahk and M. E. Zarki. Dynamic multi-path routing and how it compares with other dynamic routing algorithms for high speed wide area networks. In *Proceedings of SIGCOMM*, pages 53–64, 1992.

[4] R. Bellman. *Dynamic Prgramming*. Princeton University Press, Princeton, N.J., 1957.

[5] R. Coltun. OSPF: An Internet routing protocol. *ConneXions*, 3(8):19–25, 1989.

[6] J. A. de Azevedo, M. E. O. S. Costa, J. J. E. R. S. Madeira, and E. Q. V. Martins. An algorithm for the ranking of shortest paths. *European Journal of Operational Research*, 69:97–106, 1993.

[7] D. Eppstein. Finding the k shortest paths. In *Proc. 35th Symp. Foundations of Computer Science*, pages 154–165. Inst. of Electrical & Electronics Engineers, November 1994.

[8] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, N.J., 1962.

[9] M. L. Gardner, I. S. Loobeek, and S. N. Cohn. Type of service routing with loadsharing. In *IEEE/IECE Global Telecommunications Conference*, pages 1144–50, 1987.

[10] P. Georgatsos and D. Griffin. A management system for load balancing through adaptive routing in multiservice ATM networks. In *Proceedings of IEEE INFOCOM*, pages 863–870, 1996.

[11] J. Gray and T. McNeill. SNA multiple-system networking. IBM Systems Journal, Vol. 18, No. 2, 1979.

[12] K. Ishida, Y. Kakuda, and T. Kikuno. A routing protocol for finding two node-disjoint paths in computer networks. In *Internation Conference on Network Protocols*, pages 340–347, Nov. 1992.

[13] J. M. Jaffe, F. H. Moss, and R. A. Weingarten. SNA routing: Past, present, and possible future. *IBM Systems Journal*, 22(4):417–434, 1983.

[14] S. Keshav. *An Engineering Approach to Computer Networking*. Addison-Wesley Publishing Company, Reading, Mass., 1997.

[15] R. Krishnan and J. A. Silvester. Choice of allocation granularity in multipath source routing schemes. In *Proceedings of IEEE INFOCOM*, 1993.

[16] W. Lai. Bifurcated routing in computer networks. *Computer Communication Review*, 15(3):28–49, 1986.

[17] J. McQuillan, I. Richer, and E. C. Rosen. The new routing algorithm for the ARPANET. *IEEE Transactions on Communications*, COM-28(5):711–719, 1980.

[18] J. McQuillan and D. Walden. The ARPANET design decisions. *Computer Networks*, 1, 1977.

[19] J. Moy. The OSPF specification. RFC 1131, October 1989.

[20] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771, T.J. Watson Research Center, IBM Corp., Cisco Systems, March 1995.

[21] D. Sidhu, R. Nair, and S. Abdallah. Finding disjoint paths in networks. In *Proceedings of SIGCOMM*, pages 43–51, 1991.

RED HAT 1012
RED HAT V. CAS

[22] H. Suzuki and F. A. Tobagi. Fast bandwidth reservation scheme with multi-link & multi-path routing in ATM networks. In *Proceedings of IEEE INFOCOM*, 1992.

[23] D. M. Topkis. A k shortest path algorithm for adaptive routing in communications networks. *IEEE Transactions on Communications*, 36, 1988.

RED HAT 1012
RED HAT V. CAS

TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links

Saverio Mascolo
Politecnico Di Bari

Claudio Casetti
Politecnico Di Torino

Mario Gerla, M. Y. Sanadidi, and Ren Wang
UCLA Computer Science Department

Abstract

TCP Westwood (TCPW) is a sender-side modification of the TCP congestion window algorithm that improves upon the performance of TCP Reno in wired as well as wireless networks. The improvement is most significant in wireless networks with lossy links, since TCP Westwood relies on end-to-end bandwidth estimation to discriminate the cause of packet loss (congestion or wireless channel effect) which is a major problem in TCP Reno. An important distinguishing feature of TCP Westwood with respect to previous wireless TCP "extensions" is that it does not require inspection and/or interception of TCP packets at intermediate (proxy) nodes. Rather, it fully complies with the end-to-end TCP design principle. The key innovative idea is to continuously measure at the TCP source the rate of the connection by monitoring the rate of returning ACKs. The estimate is then used to compute congestion window and slow start threshold after a congestion episode, that is, after three duplicate acknowledgments or after a timeout. The rationale of this strategy is simple: in contrast with TCP Reno, which "blindly" halves the congestion window after three duplicate ACKs, TCP Westwood attempts to select a slow start threshold and a congestion window which are consistent with the effective bandwidth used at the time congestion is experienced. We call this mechanism *faster recovery*. The proposed mechanism is particularly effective over wireless links where sporadic losses due to radio channel problems are often misinterpreted as a symptom of congestion by current TCP schemes and thus lead to an unnecessary window reduction. Experimental studies reveal improvements in throughput performance, as well as in fairness. In addition, friendliness with TCP Reno was observed in a set of experiments showing that TCP Reno connections are not starved by TCPW connections. Most importantly, TCPW is extremely effective in mixed wired and wireless networks where throughput improvements of up to 550% are observed. Finally, TCPW performs almost as well as localized link layer approaches such as the popular Snoop scheme, without incurring the O/H of a specialized link layer protocol.

This research was supported by NSF under Grant ANI-9983138 on High Speed Networks Performance Measurements and Analysis.

1. Introduction

Effective error and congestion control for heterogeneous (wired and wireless) networks has been an active area of research recently. End-to-end, Link Layer, and Split Connection approaches have been suggested and their relative merits extensively assessed in recent studies [17]. One conclusion drawn from these studies is that while end-to-end schemes are not as effective as local recovery techniques in handling wireless losses, they are promising since significant gains can be achieved without extensive support at the network layer in routers and base stations [3]. With these motivations, we propose in this paper a TCP-based end-to-end (E2E) approach to error recovery and congestion control in wired/wireless networks and study its performance.

The well-known challenge in providing TCP congestion control in mixed environments is that current TCP implementations rely on packet loss as an indicator of network congestion. In the wired portion of the network a congested router is indeed the likely reason of packet loss; on a wireless link, on the other hand, a noisy, fading radio channel is the more likely cause of loss. This creates problems in TCP Reno since it does not possess the capability to distinguish and isolate congestion loss from wireless loss. As a consequence, TCP Reno reacts to wireless loss with a drastic reduction of the congestion window, hence of the sender transmission rate, when the best strategy would in fact be to increase the retransmission rate.

Approaches to address this problem have been discussed and compared in the excellent work by Balakrishnan et al. [3]-[5]. Three alternative approaches: E2E, Split Connection, and Localized Link Layer methods were carefully contrasted. The best performing approach was shown to be a localized link layer solution applied directly to the wireless links. A clever "snooping" protocol is introduced. The protocol appropriately called "Snoop" monitors the packets flowing over the wireless link as well as their related acknowledgments. The protocol entities cache copies of TCP data packets and monitor the ACKs in the reverse direction. If a packet loss is detected (i.e., through duplicate acknowledgments, DUPACKs), the cached copy is used for local retransmission, and any packet carrying feedback information back to the TCP sender is extracted so as to avoid "premature" retransmission at the TCP sender. The protocol is effective in reducing E2E retransmissions, and, more importantly, in preventing the associated reduction in congestion window size.

RED HAT 1012
RED HAT V. CAS

Snoop, however, has its own limitations. First, it requires a snoop proxy in the base station. Also, if the TCP sender is the mobile, the TCP code must be modified to respond to Explicit Loss Notification (ELN) packets from the base station.
In view of the limitations introduced by link layer solutions, it is of interest thus to explore E2E recovery solutions that are independent of the link layer, and thus more versatile.

In this paper we propose to handle wireless losses using a modified version of TCP Reno. This new version, which we named TCP Westwood (or TCPW for short), enhances the window control and backoff process. Namely, a TCPW sender monitors the acknowledgment reception rate and from it estimates the data packet rate currently achieved by the connection. Whenever a sender perceives a packet loss (i.e. a timeout occurs or 3 duplicate ACKs are received), the sender uses the bandwidth estimate to properly set the congestion window (cwin) and the slow start threshold (ssthresh). By backing off to cwin and ssthresh values that are based on the estimated available bandwidth (rather than simply halving the current values as Reno does), TCP Westwood avoids overly conservative reductions of cwin and ssthresh; and thus it ensures a *faster recovery*. Experimental studies reveal the benefits of the intelligent backoff strategy in TCPW: better throughput, goodput, and delay performance, as well as fairness even when competing connections differ in their end-to-end propagation times. In addition, our studies of TCPW friendliness when coexisting with TCP Reno is reassuring since we have observed that TCP Reno connections are not starved in the presence of TCPW connections. Most importantly, TCPW is very effective in handling wireless loss. This is because TCPW uses the current estimated rate as reference for resetting the congestion window. The current rate is only marginally impacted by loss (as long as loss is a relatively small fraction of data rate). The simulation results presented in Section 4 confirm this claim. For example, a throughput improvement of up to 550% over TCP Reno has been observed.

Other TCP variants that use bandwidth estimation to set the congestion window have been proposed before. To our knowledge, however, such schemes require the intervention of the network layer. For example, the BA-TCP (Bandwidth Aware-TCP) scheme [10] relies on intermediate routers to take measurements of available bandwidth and compute the "fair share" for the TCP connections. The fair share value is piggybacked in the TCP header and conveyed to the TCP source. The latter uses it to appropriately set its cwin and ssthresh parameters. BA-TCP and TCPW are similar in their reliance on bandwidth information to set congestion control parameters. However, while BA-TCP requires new network layer functions to measure available bandwidth and compute fair share, TCPW relies only on information readily available in the current TCP header. TCPW does not require any support from lower layers, and thus strictly adheres to layer separation and modularity principles.

The TCPW bandwidth estimation and congestion control algorithms are discussed in Section 2 and 3 below. TCPW performance behavior in wired and in mixed networks is studied in Sections 4 and 5. Section 6 concludes the paper.

2. End-to-End Bandwidth Measurement

2.1 The ACK-based measurement procedure

A fundamental design philosophy of the Internet TCP congestion control algorithm is that it must be performed end-to-end. The network is considered as a "black box". A TCP source cannot receive any explicit congestion feedback from the network. Therefore the source, to determine the rate at which it can transmit, must probe the path by progressively increasing the input load (through the slow start and congestion avoidance phases) until implicit feedback, such as timeouts or duplicate acknowledgments, signals that the network capacity has been reached.

The importance of the end-to-end principle [8] can never be overemphasized. In fact, it is this principle that guarantees the delivery of data over any kind of heterogeneous network.

The key idea of TCP Westwood is to exploit TCP acknowledgment packets to derive rather sophisticated measurements. We propose that a source perform an end-to-end estimate of the bandwidth available along a TCP connection by measuring and averaging the rate of returning ACKs.

After a congestion episode (i.e. the source receives three duplicate ACKs or a timeout) the source uses the measured bandwidth to properly set the congestion window and the slow start threshold, starting a procedure that we will call *faster recovery*. When an ACK is received by the source, it conveys the information that an amount of data corresponding to a specific transmitted packet was delivered to the destination. If the transmission process is not affected by losses, simply averaging the delivered data count over time yields a fair estimation of the bandwidth currently used by the source.

When duplicate ACKs (DUPACKs), indicating an out-of-sequence reception, reach the source, they should also count toward the bandwidth estimate, and a new estimate should be computed right after their reception.

However, the source is in no position to tell for sure which segment triggered the DUPACK transmission, and it is thus unable to update the data count by the size of that segment. An average of the segment sizes sent thus far in the ongoing connection should therefore be used, allowing for corrections when the next cumulative ACK is received. For the sake of simplicity, we assume all TCP segments to be of the same size.

Following this assumption, we will further assume that sequence numbers are incremented by one per segment sent, although the actual TCP implementation keeps track of the number of bytes instead: the two notations are interchangeable if all segments have the same size.

It is important to notice that, immediately after a congestion episode, followed either by a timeout or, in general, n duplicate ACKs, the bottleneck is at saturation and the connection delivery rate is equal to the share of the best-effort bandwidth (i.e., saturation bandwidth) available to that connection. At steady state, under proper conditions, as stated in Section 3.3,

RED HAT 1012
RED HAT V. CAS

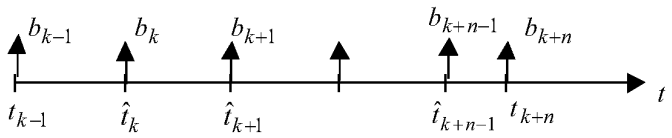


Figure.1. Bound on the maximum sampling interval obtained by inserting virtual sample

this actually should be the "fair share". The saturation condition is confirmed by the fact that packets have been dropped, an indication that one or more intermediate buffers are full. Before a congestion episode, the used bandwidth is less than or equal to the available bandwidth because the TCP source is still increasing its window to probe the network capacity.

As a result, TCP Westwood adjusts its input by taking into account the network capacity that is available to it at the time of congestion, whereas the current TCP Reno simply halves the congestion window.

2.2 Filtering the ACK reception rate.

If an ACK is received at the source at time t_k, this implies that a corresponding amount of data d_k has been received by the TCP receiver. Therefore, we can measure the following *sample* of bandwidth used by that connection as:

$$b_k = \frac{d_k}{t_k - t_{k-1}}$$

where t_{k-1} is the time the previous ACK was received. Since congestion occurs whenever the low-frequency input traffic rate exceeds the link capacity [15], we employ a low-pass filter to average sampled measurements and to obtain the low-frequency components of the available bandwidth. Notice that this averaging is also critical to filter out the noise due to delayed acknowledgments.

The choice of the filter is important. According to our experience, simple exponential averaging of the kind used by TCP for RTT estimation is unable to efficiently filter out high-frequency components of the bandwidth measurements. We propose the following discrete time filter which is obtained by discretizing a continuous low-pass filter using the Tustin approximation [19] We obtain:

$$\hat{b}_k = \frac{\frac{2\tau}{t_k - t_{k-1}} - 1}{\frac{2\tau}{t_k - t_{k-1}} + 1}\hat{b}_{k-1} + \frac{b_k + b_{k-1}}{\frac{2\tau}{t_k - t_{k-1}} + 1} \quad (1)$$

where $\hat{b}_k$ is the filtered measurement of the available bandwidth at time $t = t_k$, and $1/\tau$ is the cut-off frequency of the filter. The structure of filter (1) is simple. To understand how it works it is useful to consider a constant interarrival time $t_k - t_{k-1} = \Delta_k = \tau/10$. Then, filter (1) becomes a filter with constant coefficients

$$\hat{b}_k = a\hat{b}_{k-1} + \frac{(1-a)}{2}[b_k + b_{k-1}] \quad (2)$$

where $a = 0.9$. The particular form (2) shows that the new value $\hat{b}_k$ is made by the 90% of the previous value $\hat{b}_{k-1}$ plus the 10% of the arithmetic average of the last two sample b_k and b_{k-1}. Even though the filter (2) is useful to explain how the average is computed, it cannot be used because in packet-switched networks the interarrival time is not constant. To counteract the effect of time-varying interarrival time, the coefficients of filter (1) depend on $t_k - t_{k-1}$. The effect of using time-varying coefficients is easy to understand. When the interarrival time increases, the last value $\hat{b}_{k-1}$ should have less significance, since it represents an older value, whereas the significance of recent samples should be higher. This is exactly what happens with filter (1): the a coefficient decreases when the interarrival time increases meaning that the previous value $\hat{b}_{k-1}$ has less significance with respect to the last two recent samples which are multiplied by $(1-a)$.

Finally, filter (1) has a cut-off frequency equal to $1/\tau$. This means that all frequency components above $1/\tau$ are filtered out. According to the Nyquist sampling theorem, in order to sample a signal with bandwidth $1/\tau$ a sampling interval less or equal to $\tau/2$ is necessary. But, since the ACK stream may be irregular (for instance, no ACKs are returned when the sender is idle), the sampling frequency constraint cannot be guaranteed. To guarantee the Nyquist constraint and thus preserve the low pass filter effect, we establish that if a time τ/m ($m \geq 2$) has elapsed since the last received ACK without receiving any new ACK, then the filter assumes the reception of a *virtual* sample $b_k=0$. The situation is shown in Fig. 1 below, where t_{k-1} is the time an ACK is received, $\hat{t}_{k+j}$ are the arrival times of the virtual samples, with $\hat{t}_{k+j+1} - \hat{t}_{k+j} = \tau/m$ for $j=0,n-1$, $b_{k+j} = 0$ for $j=0,n-1$ are the virtual samples and $b_{k+n} = \frac{d_{k+n}}{t_{k+n} - t_{k+n-1}}$ is the sample computed when the ACK is received at t_{k+n}.

It is interesting to look at the form that filter (1) takes in the persistent absence of ACKs from $t=t_k$: that is, in the absence of ACKs the estimated bandwidth exponentially goes to zero.

$$\hat{b}_k = \frac{2m-1}{2m+1}\hat{b}_{k-1} + \frac{0 + b_{k-1}}{2m+1}$$

$$\hat{b}_{k+1} = \frac{2m-1}{2m+1}\hat{b}_k$$

.....

RED HAT 1012
RED HAT V. CAS

$$\hat{b}_{k+h} = \left(\frac{2m-1}{2m+1}\right)^h \hat{b}_k$$

In our experiments, m was set to 2.

2.3 On the effects of delayed and cumulative ACKs on bandwidth measurement

As previously stated, DUPACKs should count toward the bandwidth estimation, since their arrival indicates a successfully received segment, albeit in the wrong order. As a consequence, a cumulative ACK should only count as one segment's worth of data since duplicate ACKs ought to have already been taken into account. However, the matter is further complicated by the issue of *delayed ACKs*. The standard TCP implementation provides for an ACK being sent back once every other in-sequence segment received, or if a 200-ms timeout expires after the reception of a single segment [18]. The combination of delayed and cumulative ACKs can potentially disrupt the bandwidth estimation process.

We therefore stress two important aspects of the bandwidth estimation process:

a. The source must keep track of the number of DUPACKs it has received before new data is acknowledged;
b. The source should be able to detect delayed ACKs and act accordingly.

The approach we have chosen to take care of these two issues can be found in the *AckedCount* procedure, detailed below, showing the set of actions to be undertaken upon the reception of an ACK, for a correct determination of the number of packets that should be accounted for by the bandwidth estimation procedure, indicated by the variable *acked* in the pseudocode. The key variable is *accounted*, which keeps track of the received DUPACKs. When an ACK is received, the number of segments it acknowledges is first determined (*cumul_ack*). If *cumul_ack* is equal to 0, then the received ACK is clearly a DUPACK and counts as 1 segment towards the BWE; the DUPACK count is also updated. If *cumul_ack* is larger than 1, the received ACK is either a delayed ACK or a cumulative ACK following a retransmission event; in that case, the number of ACKed segments is to be checked against the number of segments already accounted for (*accounted_for*). If the received ACK acknowledges fewer or the same number of segments than expected, it means that the "missing" segments were already accounted for when DUPACKs were received, and they should not be counted twice. If the received ACK acknowledges more segments than expected, it means that although part of them were already accounted for by way of DUPACKs, the rest are cumulatively acknowledged by the current ACK; therefore, the current ACK should only count as the cumulatively acknowledged segments. It should be noted that the last condition correctly estimates the delayed ACKs (*cumul_ack* = 2 and *accounted_for* = 0).

```
PROCEDURE AckedCount

cumul_ack      =      current_ack_seqno      -
last_ack_seqno;
if (cumul_ack = 0)
  accounted_for=accounted_for+1;
  cumul_ack=1;
endif

if (cumul_ack > 1)
  if (accounted_for >= cumul_ack)
    accounted_for=accounted_for-cumul_ack;
    cumul_ack=1;
  else if (accounted_for < cumul_ack)
         cumul_ack=cumul_ack-
accounted_for;
         accounted_for=0;
  endif
endif

last_ack_seqno=current_ack_seqno;
acked=cumul_ack;

return(acked);

END PROCEDURE
```

3. TCP Westwood

In this Section we describe how the bandwidth estimation can be used by the congestion control algorithm executed at the sender side of a TCP connection. As will be explained, the congestion window dynamics during slow start and congestion avoidance are unchanged, that is they increase exponentially and linearly, respectively, as in current TCP Reno.

The general idea is to use the bandwidth estimate *BWE* to set the congestion window (*cwin*) and the slow start threshold (*ssthresh*) after a congestion episode.

We start by describing the general algorithm behavior after n duplicate ACKs and after coarse timeout expiration.

3.1 Algorithm after n duplicate ACKs

The pseudocode of the algorithm is the following:

```
if (n DUPACKs are received)

  ssthresh =  (BWE*RTTmin)/seg_size;
  if (cwin>ssthresh) /* congestion avoid.
*/
    cwin = ssthresh;
  endif

endif
```

Note that *seg_size* identifies the length of a TCP segment in bits.

During the congestion avoidance phase we are probing for extra available bandwidth. Therefore, when n DUPACKs are received, it means that we have hit the network capacity (or that, in the case of wireless links, one or more segments were dropped due to sporadic losses). Thus, the slow start threshold is set equal to the available pipe size when the bottleneck buffer is empty, which is *BWE*RTTmin*, the congestion window is set equal to the *ssthresh* and the congestion avoidance phase is entered again to gently probe for new available bandwidth. The

RED HAT 1012
RED HAT V. CAS

value *RTTmin* is set as the smallest RTT sample observed over the duration of the connection. This setting allows the queue be drained after a congestion episode. During the slow-start phase we are still probing for the available bandwidth. Therefore the BWE we obtain after *n* duplicate ACKs is used to set the slow start threshold. After *ssthresh* has been set, the congestion window is set equal to the slow start threshold only if *cwin>ssthresh*. In other words, during slow start, *cwin* still features an exponential increase as in the current implementation of TCP Reno.

3.2 Algorithm after coarse timeout expiration

The pseudocode of the algorithm is

```
if (coarse timeout expires)
  ssthresh = (BWE*RTTmin)/seg_size;

  if (ssthresh < 2)
    ssthresh = 2;
  endif;

  cwin = 1;
endif
```

The rationale of the algorithm is again simple. After a timeout *cwin* and *ssthresh* are set equal to 1 and BWE, respectively, so that the basic Reno behavior is still captured, while a speedy recovery is granted by the ssthresh being set to the bandwidth estimation at the time of timeout expiration.

3.3 TCP Westwood convergence to fair share

An important goal of any TCP implementation is for every connection to get its "fair share" of the bottleneck. We will use an informal argument similar to that used for Reno in [14] to show that TCPW achieves the fair share. Consider the case of two connections with the same RTTs. Suppose, for the sake of example, that the RTT is X packet transmission times, and the bottleneck has X buffers. One connection, say A, starts first. Its window "cycles" between X and 2X (as per the TCPW algorithm described earlier in this Section), each cycle terminating when buffer overflows. Later, connection B starts, first in slow start mode, and then in congestion avoidance mode. In congestion avoidance, during each cycle the windows A and B grow approximately at the same rate, i.e., one segment per RTT. Eventually, the bottleneck buffer overflows, terminating the cycle. One can show that the window at overflow is:

$W_i = R_i (b/C + RTT)$, for i = A,B;

where R is the achieved rate (i.e., BWE); b is the bottleneck buffer size; and C is the bottleneck trunk capacity.

This is a general property true for all TCP protocols, and in particular for TCPW. After overflow, TCPW reduces the windows to new values W_i' as follows:

$W_i' = R_i (RTT)$ for i = A, B

Thus, the ratios of the windows A and B are preserved after overflow. Yet, the ratio W_B/W_A keeps increasing during congestion avoidance. Consequently, the B window and throughput ratchet up at each cycle. Equilibrium is reached when the two connections have the same windows and the same bandwidth fair share. Figure 2 graphically illustrates the convergence to the fix point $W_A=W_B$.

This informal proof is validated by actual simulation results. It can be generalized to many simultaneous connections (all with the same RTT). It can also be applied to the case when the bottleneck is affected by random errors equally hitting all connections.

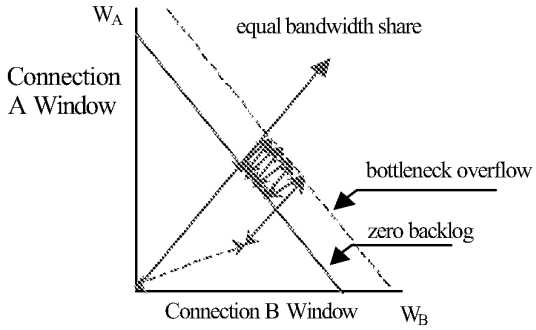


Figure 2. Convergence toward fair bandwidth sharing.

4. TCP Westwood Performance, Fairness, and Friendliness

In this Section, we report on the basic performance behavior of TCPW, its fairness among a number of TCPW connections sharing a bottleneck link, and its friendliness to coexisting connections of other TCP variants, such as Reno.

First, the effectiveness of the bandwidth estimation algorithm is studied using a single TCP connection and a fluctuating UDP traffic rate. TCPW window dynamics (*cwin*, *ssthresh* and sequence numbers) are then considered. TCPW performance behavior is compared to the standard and widely used TCP Reno as well as to TCP SACK [16].

All simulations presented in this paper were run using the LBL network simulator, 'ns' ver.2 [19].

New simulation modules for TCP Westwood were written and they are available at [20], while existing modules for simulations involving TCP Reno and TCP Sack were used. All simulated TCP receivers implement delayed-ACKs. Notice that this introduces a complication for our bandwidth estimation algorithm as delayed ACKs represent noise to be filtered, as explained in Section 2.3.

Each scenario, involving different bottleneck link capacity, RTT or number of concurrent connections, includes a single-bottleneck link as is common in the literature. Intermediate node buffer capacity is always set equal to the bandwidth-delay product for the scenario under study. The packet size is set to 400 bytes in all experiments. The ACK arrival pattern is

RED HAT 1012
RED HAT V. CAS

repetitive for each RTT in absence of packet losses (errors or buffer overflow). Thus, the interval τ should span one or more RTTs. Experimentally, we have observed that performance is not very sensitive to the choice of τ as long as τ>RTT. In our experiments, we set τ equal to 500ms.

4.1 Bandwidth estimation effectiveness

In this Section, we test the effectiveness of the proposed bandwidth estimation algorithm. For this purpose we consider a single TCPW connection sharing the bottleneck link with UDP connections. Packets are queued and transmitted on the link in FCFS order. In addition to demonstrating the accuracy of the bandwidth estimation algorithm, this scenario also illustrates the capability of a TCP Westwood connection to use the bandwidth left over by dynamic UDP flows. The configuration simulated here features a 5 Mb/s bottleneck link with a one-way propagation delay of 30ms. One TCP connection shares the bottleneck link with two ON/OFF UDP connections, and TCP and UDP packets are assigned the same priority. Each UDP connection transmits at a constant bit rate of 1 Mb/s while ON. Both UDP connections start in the OFF state; after 25s, the first UDP connection is turned ON, joined by the second one at 50s; the second connection follows an OFF-ON-OFF pattern at times 75s, 125s and 175s; at time 200s the first UDP connection is turned off as well. The UDP connections remain silent until the end of the simulation. The TCPW connection sends data throughout the simulation.

The scenario above is intended to demonstrate the effectiveness of the feedback control used in TCPW when subjected to "step" and "impulse" stimuli. The behavior of the bandwidth estimation process is shown in fig. 3.

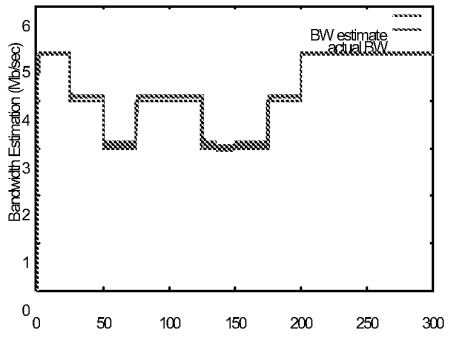


Figure 3. TCPW with concurrent UDP traffic: bandwidth estimation

4.2 TCPW fairness

Fair bandwidth sharing implies that all connections are provided with similar opportunity to transfer data. Our experiments show that TCPW fairness is at least as good, if not better, than that provided by the widely-used TCP Reno. In the sample results below we show that two flows with different E2E round trip times (RTT) share the bandwidth more effectively under TCPW than under TCP Reno.

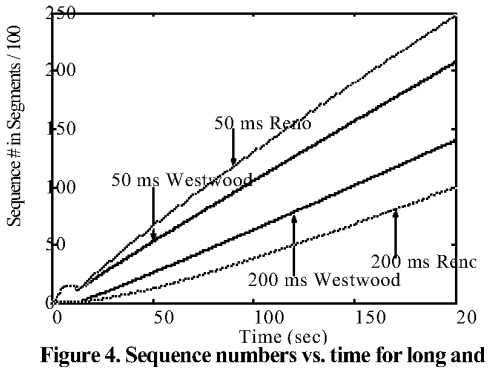


Figure 4. Sequence numbers vs. time for long and short RTT connections without RED

We ran simulations in which connections were subject to 50ms and 200ms RTT, respectively. Figure 4 and Figure 5 show sequence number progress for TCPW and Reno connections without and with RED, respectively. In all cases the short connection progresses faster as expected. We note however that TCPW provides better fairness than Reno across different propagation times. The reason for the superior fairness exhibited by TCPW is that the long connection suffers less reduction in *cwin* and *ssthresh*. In Reno, *cwin* reduction is independent of RTT. The results in Figure 5 show that both protocols benefit from RED, as far as fairness is concerned. Remarkably, the improvement in TCPW due to RED was higher than the improvement in Reno.

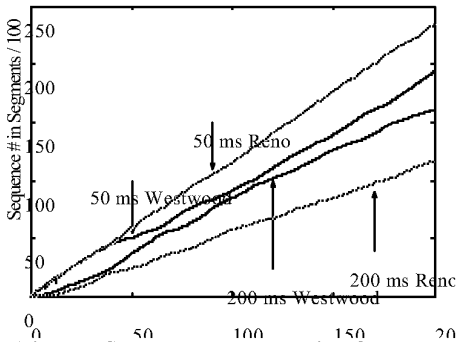


Figure 5. Sequence numbers vs. time for long and short RTT connections with RED

4.3 TCPW friendliness

Friendliness is another important property of a TCP protocol. TCPW must be "friendly" to other TCP variants. That is, TCPW connections must be able to coexist with connections from TCP variants while providing opportunities for all connections to progress satisfactorily. At least, TCPW connections should not result in starvation of connections running other TCP variants. Better yet, the bandwidth share of TCPW connections should be equal to their fair share.

We ran simulation experiments with the following parameters: 2-Mbps bottleneck link, 20 flows total, all flows with 100ms RTT. With all 20 connections running TCPW, the average throughput per connection was 0.0994 Mbps. All 20 Reno

RED HAT 1012
RED HAT V. CAS

connections resulted in an average throughput of 0.0992 Mbps. As predicted, we got the same results for the two schemes. We then ran 10 Reno with 10 Westwood connections sharing the same 2Mbps bottleneck link over a path of 100ms RTT. The average throughput for a TCPW connection went up to 0.1078, and that of a Reno connection went down to 0.0913. This shows that TCPW behavior departs from "fair share" by 16% (TCPW gains 8% and TCP Reno loses 8%). This unfairness is rather moderate and it can be tolerated as it allows for coexistence with Reno.

To probe the friendliness issue further, we also carried out actual measurements using our TCPW Linux implementation in our lab. Figure 6 shows the topology of our lab test bed. The link emulator is used to vary the link propagation time and error characteristics.

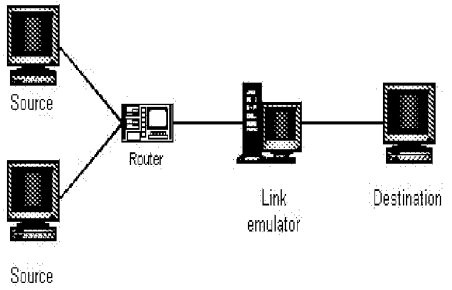


Figure 6. Experimental test bed layout

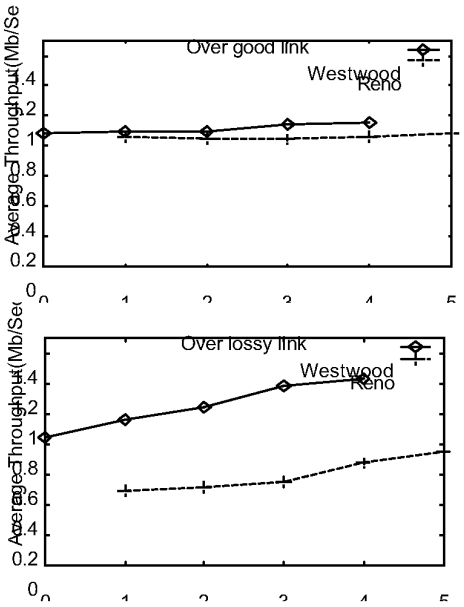


Figure 7. Average throughput vs. No. of Reno connections over good and lossy link (5 connections total)

We measured the throughput for a total of 5 connections with a variable Reno/ TCPW mix. Then, to evaluate the friendliness of TCPW under stress, we introduce a relatively high error rate on the bottleneck link, namely 1% packet loss (See Figure 7). This error rate is actually appropriate for wireless links as we shall discuss later. Note that TCP West shines in presence of line errors, so, friendliness in the error situation is even more difficult to establish than in error free operation.

The RTT was 100ms. Error rates and propagation delays are introduced in our test bed via a link/network emulator. Figure 7 shows the average throughput per connection for TCPW and for Reno. The lower average throughput line is that of the Reno connections. The horizontal axis represent the number of Reno connections in the mix. For example, at the point marked 3 on the horizontal line, the measurement experiment includes 3 Reno connections and 2 TCPW connections The results in Figure 7 illustrate two important points. First, TCPW has a significant edge in a high-error-rate environment: 5 TCPW get 10% more throughput than 5 TCP Reno. We will press more on this later. Secondly, friendliness is preserved. Even though TCPW has an advantage over Reno in error-prone environments, Reno connections were not starved. In fact, the introduction of TCPW connections into the mix reduces the average throughput of a Reno connection only by a minimal amount. Thus, for practical purposes, we can claim that TCPW is friendly.

5. TCPW Performance In Mixed (Wired and Wireless) Networks

TCPW is being proposed in this paper as an end-to-end solution to error and congestion control in mixed wired and wireless networks. In view of this claim, a number of different scenarios are studied below to show the benefits of using TCPW in such wired/wireless environments. Independent and correlated loss models are used. Ground radio as well as satellite scenarios are developed and studied.

5.1 Independent loss model in ground radio environment

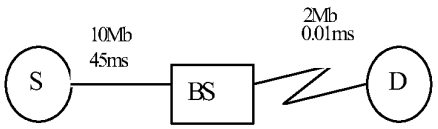


Figure 8. A simple Simulation Topology

Figure 8 shows a topology of a mixed network with a wired portion including a 10-Mbps link between a source node and a base station. The propagation time over the wired link is initially assumed to be 45ms. Later, the propagation time is varied from 0 to 250ms to represent a variety of wired network environments ranging from campus to intercontinental connections.

RED HAT 1012
RED HAT V. CAS

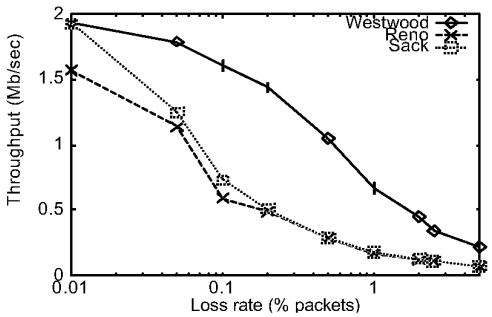


Figure 9. Throughput vs. error rate of the wireless link

The wireless portion of the network is a very short 2Mbps wireless link with a propagation time of 0.01ms. The wireless link is assumed to connect the base station to a destination mobile terminal. Errors are assumed to occur in both directions of the wireless link.

We compare the throughput of TCPW to that of Reno and SACK assuming independent (Bernoulli) errors ranging from 0 to 5% packet loss probability. The error model assumed here is equivalent to the "exponential error" model in which the time between successive errors is exponentially distributed [3]. The range of error rates assumed here is also similar to the range used in [3]. The results in Figure 9 show that TCPW gains up to 394 % over Reno or SACK. This gain occurs at a realistic packet error probability of 1%.

To assess TCPW throughput gain and its relation to the E2E propagation time, we ran simulations with the wired portion propagation time varying from 0 to 250ms. The results in Figure 10 show a significant gain for TCPW of up to 567%, at a propagation time of 100ms. When the propagation time is small (say, less than 5ms), all protocols are equally effective. This is because a small window is adequate and window optimization is not an issue. TCPW reaches maximum improvement over Reno and Sack as the propagation time increases to about 100ms. After that, in this experiment, the gain starts to decrease as the feedback information used to estimate the available bandwidth arrives too late to be of significant help to TCPW.

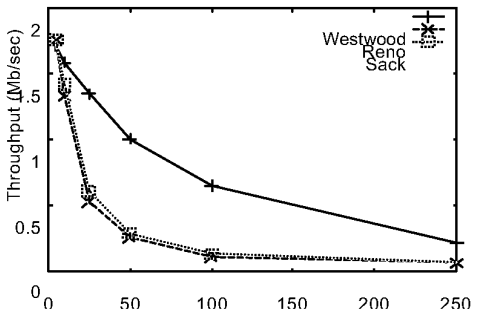


Figure 10. Throughput vs. one-way propagation delay

Simulation results in Figure 11 show that TCPW gains also increase significantly as the bottleneck link transmission speed increases (again, because what matters is the window size determined by the bandwidth-delay product). Thus, TCPW is more effective than TCP Reno in utilizing the Gbps bandwidth provided by new-generation, high-speed networks. Figure 11 shows that the improvement obtained via TCPW increases to approximately 550% when the wireless link speed reaches 8 Mbps. The error model is still Bernoulli with parameter 0.5%, and the E2E propagation time is 45ms.

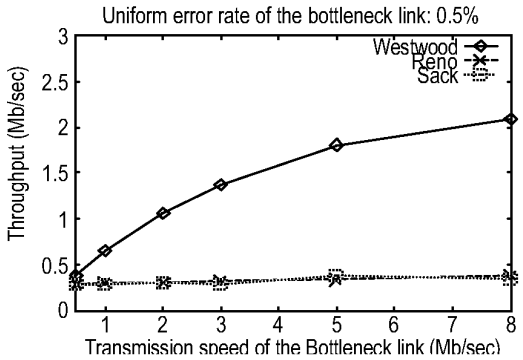


Figure 11 Throughput vs. link capacity

Window dynamics of TCPW and of TCP Reno are presented in Figure 12 and Figure 13 below. The graphs show the improved window dynamics in TCPW. The cwin and ssthresh values are consistently higher than the corresponding values in Reno, thus yielding higher throughput .

Next we compare TCPW to Snoop, the leading local strategy shown to provide the biggest improvement over TCP Reno [4]. Published results show that Snoop provides approximately a 400% improvement over an E2E approach based on TCP Reno when the error rate is 1 bit in 64 KBytes and the round trip propagation time is 135ms. Our simulations with similar parameter values show that TCPW provides a 382%

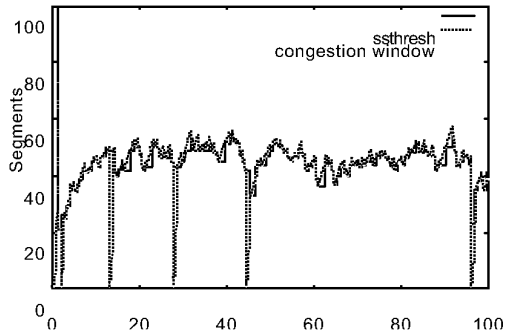


Figure12. TCP Westwood over lossy link—cwin and ssthresh

RED HAT 1012
RED HAT V. CAS

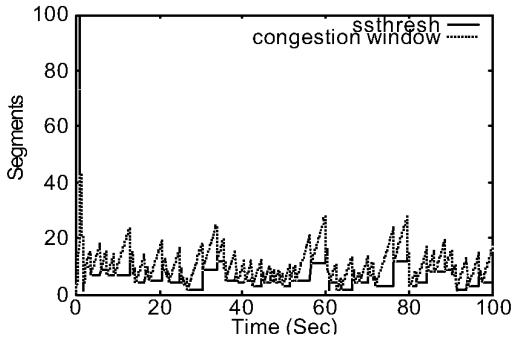


Figure 13. TCP Reno over lossy link—cwin and ssthresh

improvement over Reno. This shows that TCPW and Snoop gains are remarkably (and enticingly) close. We plan to probe further the issue of effectiveness of local versus E2E error recovery via simulation and measurements. From the qualitative and protocol implementation standpoint, however, we note that TCPW is completely end-to-end, and does not require any support from network or link layers. It does not have the scalability problems that Snoop may encounter as the number of mobile terminals increase. Further, the effectiveness of Snoop in wireless subnets including multiple base stations and handoffs is not clear.

Explicit Loss Notification (ELN) is an E2E scheme that is introduced and assessed in [3]. Basically, the method provides explicit notification from TCP receiver to TCP sender that a loss *due to a link error* has occurred. The lost packet is also identified to the Sender TCP entity. Using the same parameter values above (1 bit in 64 Kbytes error rate, and 135ms propagation time), ELN is shown to provide a gain of approximately 200% over Reno. In comparison, TCPW provides 382%, closer to Snoop performance. Further, ELN assumes that the destinations can detect errors on a link and identify the packet and its TCP source. These assumptions are not likely to be uniformly satisfied for various error causes and various link technologies, thus the limitation in versatility of ELN in addition to its limited gain over Reno.

We compared, via simulation, TCPW to BA-TCP [10], an alternative strategy where the routers explicitly measure and relay the bandwidth available for each connection back to the TCP sender. At 40ms round trip time, and 1 bit in 100KB error rate, BA-TCP's improvement over Reno is 202%. For TCP Westwood, the throughput improvement at the same parameter values is 161%. This is quite remarkable considering that BA-TCP measures the bandwidth actually available for a connection at the bottleneck router, while TCPW works with no support from routers to estimate the available bandwidth at the bottleneck. Note that the router functionalities required by BA-TCP are not available in today's routers.

5.2 Burst error models in a ground radio environment

To study TCPW performance with correlated errors, we use the 2-state Markov models following [1][5]. In such models, burst errors occur at a high rate due to a variety of conditions associated mostly with terminal mobility. Such conditions include variable fading, blackouts due to shadowing, and the like. Figure 14 below depicts the 2-state Markov model. The wireless link is assumed to be in one of two states: Good or Bad. In the Good state, a bit (or packet) error Bernoulli model is assumed. The time intervals between bit errors is thus exponentially distributed (memoryless channel errors). In addition, a link is assumed to stay in the Good state a time interval that is exponentially distributed with parameter λ_{gb} . The time spent in the Bad state is also exponentially distributed but with parameter λ_{bg}. In the Bad state we assume that errors are still Bernoulli, however, the rates of errors in the Bad state are much higher. For the simulation experiments below we vary the error rate in the bad state depending on the specific link conditions we want to study. To represent fading conditions, the bit error rate is assumed to range from 0 to 30%. For blackouts, the error rate is 100%.

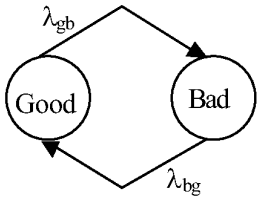


Figure 14. 2-state Markov Model for Burst Error Characterization

Simulation results using the 2-state Markov models show that TCPW improves throughput for links with fading and blackouts as discussed below.

(a) Fading:

Let the Bad state represent fading conditions, and let the mean duration of Good and Bad states be 8 and 4 sec respectively. The error rate in the Good state is assumed to be 0.001% packet loss, and the error rate in the Bad state is varied from 0 to 30% packet loss rate. The results in Figure 15 show the improvement obtained with TCPW over Reno or SACK. TCPW increases throughput by up to 300%. This is achieved when the error rate in the Bad state is 5 %. When the error rate is higher, all protocols perform poorly.

When the error rate is less than 5%, TCPW provides about 150% improvement. We also varied the link speed to determine its impact on the protocols performance.

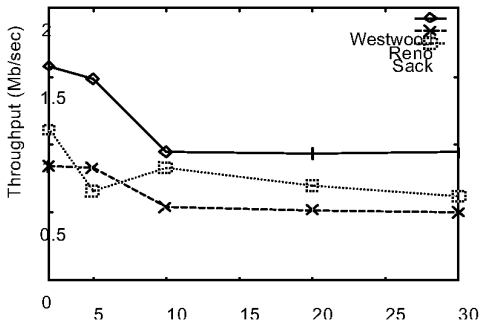


Figure 15. Throughput vs. error rate of the bad state

RED HAT 1012
RED HAT V. CAS

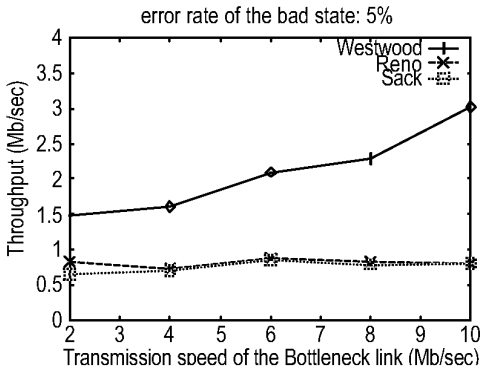


Figure. 16. Throughput vs. link capacity in 2-state error model

Figure 16 shows that TCPW improvement increases as the wireless link (bottleneck link in this case) increases (as expected, since a similar trend was observed also in wired links). At 10 Mbps link speed, a 400% throughput improvement is achieved at the same error rate in Bad state of 5% packet loss.

(b) Blackouts:

Let us now assume that the Bad state represents a blackout, where a base station becomes temporarily not visible to a terminal due to mobility. The mean duration for the Good state is 4 sec; the mean duration of the Bad state varies between 0 and 0.5 sec, Figure 17 shows the throughput improvement obtained by TCPW to be 167 % over Reno and SACK when the mean blackout duration is 0.1 sec. For longer blackouts, TCP timeouts occur and all protocols are equally affected.

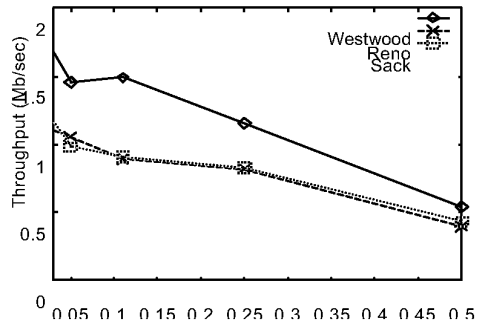


Figure 17 Throughput vs. average duration of blackout

5.3 LEO Satellite Model

Another environment where TCPW is likely to be valuable is the LEO satellite system. LEO Satellites present an environment with varying link quality and relatively long propagation delay. Also, in the future, higher transmission speeds are expected. That is where TCPW would be most beneficial.

We considered for this study a scenario where a single hop, up to the satellite and down to an earth terminal, connects a terminal to a gateway and from there to the terrestrial network. One way (e.g. terminal to gateway) propagation time is assumed to be 100ms. The error rate is assumed 0.1% in normal operating conditions. Occasionally, if the LEO system supports satellite diversity, a handoff to a different LEO satellite (different orbit) becomes necessary to overcome the blocking due to buildings, thick foliage etc. During handoff, we assume all packets are lost. In our model, the handoff from one satellite to another needs 100ms to complete, and the period between handoffs is 4s, say. Figure 18 below shows the major improvements of TCPW over Reno and SACK, especially at high speeds.

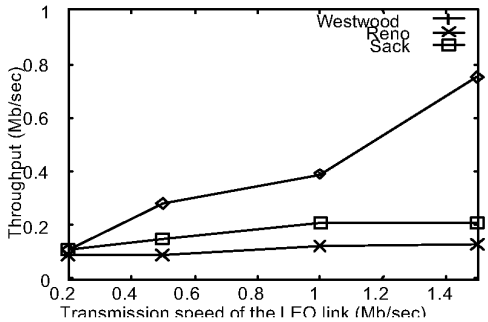


Figure 18. Throughput vs. link capacity of the Satellite link

6. Conclusions and Future Research

In this paper we have proposed a new version of the TCP protocol, TCP Westwood, (TCPW for short) aimed at improving performance under random or sporadic losses. TCPW has been tested through simulation, showing considerable throughput gains in almost all wireless scenarios.

In retrospect, the new scheme can be viewed as one more step in the TCP evolution. TCP Tahoe resets *cwin* to one after a loss. TCP Reno halves *cwin* after three duplicate ACKs. TCP Westwood introduces a "faster" recovery mechanism to avoid over-shrinking *cwin* after three duplicate ACKs. It does so by taking into account the end-to-end estimation of the bandwidth available to TCP. The use of bandwidth estimation feedback to control the congestion window has an effect that goes beyond faster recovery. Namely, TCP window congestion control is based not solely on packet loss (which itself is an ambiguous congestion indicator in presence of wireless links), but also on the bandwidth estimate at the time of loss. The benefits of using bandwidth estimation (in addition to packet loss) have been amply demonstrated in a very broad range of wireless scenarios.

The issue of friendliness, raised by previous reviewers of this work, has been addressed. A qualitative proof of fair behavior under appropriate conditions has been provided. "Unfriendly" trends due to TCPW "aggressiveness" have been detected in our experiments, but were shown to be contained and never severe enough to lead to starvation.

The code modifications required to implement TCP Westwood are comparable to the ones implemented in the transition from TCP Tahoe to TCP Reno. As in the Tahoe to Reno transition, a

RED HAT 1012
RED HAT V. CAS

major advantage of the TCP Westwood modification is that it affects only the source TCP (as opposed to other variants such as TCP SACK that require also destination modifications). This allows a TCP Westwood source to interwork with arbitray destinations in the Internet.

Work is in progress in many directions. Some of these directions were indicated by the anonymous referees – we thank them for their thorough reviews and valuable comments. We are planning to include in TCPW the NewReno feature that allows efficient recovery from multiple losses in the same window. We are aware that in some cases the bottleneck link is in the backward path, from receiver to sender. In such cases, the bottleneck must be "fairly" shared among Data Packets (in some connections) and ACKs (in some other connections). We plan to attack this problem by defining first a suitable measure of "fairness" between Data and ACK streams. If fairness is defined as equal throughput for all connections regardless whether Data or ACK bound, and Data packet size is the same for all connections, one can show that TCPW provides a fair solution – at equilibrium all connections measure the same BWE. The comparison of TCPW with link level techniques such as Snoop deserves further study. It is clear that link level recovery is in general much more powerful than end to end recovery since it isolates and corrects the loss "in loco". For instance, suppose that the bottleneck is in the wired network and one of the connections sharing the bottleneck goes over a wireless, lossy link. With E2E recovery (TCPW and TCP Reno alike) the wireless connection is heavily penalized with respect to the others. With link layer recovery (eg, Snoop) fair sharing is enforced. Next, TCPW performs poorly when random packet loss rate exceeds a few percent. Snoop, on the other hand, is quite robust to high error rates. We are now investigating TCPW enhancements that will in part correct these deficiencies.. We plan to further refine our bandwidth estimation and filtering method, in order to improve TCPW "friendliness". Finally, we intend to pursue the development of control theoretical models that will enable us to study the stability of TCPW as a function of the various systems parameters.

7. Acknowledgments

The authors take great pleasure in acknowledging the valuable contribution to this work by Tienshiao Ma, Bryan Ng, Giovanni Pau and Cathy Yang. who valiantly implemented TCPW under Linux and conducted the lab measurements experiments reported above.

8. References

[1] Abouzeid, A.A, S. Roy, M. Azizoglu. Stochastic Modeling of TCP over Lossy link INFOCOM 2000, Tel Aviv, Israel, March 2000.

[2] Åström ,K. J., B. Wittenmark, Computer controlled systems, Prentice Hall, Englewood Cliffs, N. J., 1997].

[3] Balakrishnan, H., V. N. Padmanabhan, S. Seshan, and R. H. Katz,. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. IEEE/ACM Transactions on Networking, December 1997.

[4] Balakrishnan, H., S. Seshan, E. Amir and R. H. katz. Improving TCP/IP Performance Over Wireless Networks. MOBICOM'95, Berkeley, CA, USA, November 1995

[5] Balakrishnan, H., and Randy H. Katz. Explicit Loss Notification and Wireless Web Performance. Proc. IEEE GLOBECOM'98 Internet Mini-Conference, Sydney, Australia, November 1998.

[6] Bonald, T., Comparison of TCP Reno and TCP Vegas: Efficiency and Fairness. In Proceedings of PERFORMANCE'99, Istanbul, Turkey, October 1999.

[7] Casetti, C., M. Gerla, S. Lee, S Mascolo and M. Sanadidi, "TCP with Faster Recovery", MILCOM 2000, Los Ange les, CA, October 2000.

[8] Clark, D., "The design philosophy of the DARPA Internet protocols", Proc. of Sigcomm88 in ACM Computer Communication Review, vol. 18, no. 4, pp. 106-114, 1988.

[9] Gerla, M., R. Lo Cigno, S. Mascolo, and W. Weng. Generalized Wind ow Advertising for TCP Congestion Control. CSD-TR 990012, UCLA, CA, USA, February 1999.

[10] Gerla, M., W. Weng and R. Cigno, "Bandwidth feedback control of TCP and real time sources in the Internet", GLOBECOM'2000, San Francisco, CA, USA, November 2000.

[11] T. Henderson, Satellite Transport Protocol Specification. Technical Report, University of California, Berkeley, 1999.

[12] Hengartner, U., J. Bolliger, and T. Gross. TCP Vegas Revisited. In Proceedings of IEEE INFOCOM'2000, Tel Aviv, Israel, March 2000.

[13] Jacobson, V., Congestion Avoidance and Control. ACM Computer Communications Review, 18(4):314--329, August 1988.

[14] Kurose, J., and K. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", Addison Wesley, 2000

[15] Li, S. Q. and C. Hwang, "Link Capacity Allocation and Network Control by Filtered Input Rate in High speed Networks", IEEE/ACM Transections on Networking, vol. 3, no. 1, Feb. 1995, pp. 10-25

[16] Mathis, M., J. Mahdavi, S.Floyd, and A.Romanow. TCP Selective Acknowledgemwnt Options. RFC 2018, April 1996

[17] Mitzel, D., Overviw of 2000 IAB Wireless Internetworking Workshop, RFC 3002.

[18] Stevens, W.R. TCP/IP Illustrated, vol. 1. Addison Wesley, Reading, MA, USA, 1994

[19] ns-2 network simulator (ver 2). LBL, URL: http://www-mash.cs.berkeley.edu/ns.

[20] TCP Westwood modules for ns-2. URL: http://www.telematics.polito.it/casetti/tcp-westwood.

RED HAT 1012
RED HAT V. CAS

Electronic Acknowledgement Receipt

EFS ID:	19203299
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	03-JUN-2014
Filing Date:	22-JUN-2012
Time Stamp:	21:32:14
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Form (SB08)	13531294_IDS_6-3-2014.pdf	613644 53d223d9719b37ed817be102a61072535d1bc5b9	no	5
Warnings:					
Information:					

RED HAT 1012
RED HAT V. CAS

A U.S. Patent Number Citation or a U.S. Publication Number Citation is required in the Information Disclosure Statement (IDS) form for autoloading of data into USPTO systems. You may remove the form to add the required data in order to correct the Informational Message if you are citing U.S. References. If you chose not to include U.S. References, the image of the form will be processed and be made available within the Image File Wrapper (IFW) system. However, no data will be extracted from this form. Any additional data such as Foreign Patent Documents or Non Patent Literature will be manually reviewed and keyed into USPTO systems.

2	Transmittal Letter	13531294_IDS_Transmittal_Letter.pdf	72566 97d17281196ae6f4b16baafea91a69002b37a4ee	no	1
Warnings:					
Information:					
3	Non Patent Literature	Chen_Thesis_1999.pdf	10797452 2b7566f7001fe6ed5de8cf82e1ecdd22667842a3	no	258
Warnings:					
Information:					
4	Non Patent Literature	Hsieh_Mobicom_Sept-2002.pdf	295997 836e5ac81b478d54535976116655c27235071565	no	12
Warnings:					
Information:					
5	Non Patent Literature	Kohler_ACM_2000.pdf	528647 6baec406fd642672ffb0a3ebb4fa49b1e6480f69	no	34
Warnings:					
Information:					
6	Non Patent Literature	Lee_Infocom_1998.pdf	118051 4f5abdf6655110e8873f662cc4baa0a9902fe024	no	5
Warnings:					
Information:					
7	Non Patent Literature	Lee_JCN_6-2002.pdf	155046 2590a7b751e40d5e79c27a6617a6852292ff3167	no	10
Warnings:					
Information:					
8	Non Patent Literature	Magalhaes_ICNP-2001.pdf	751363 62077d5586cd055865abfc4882757e3ad5378b04	no	7
Warnings:					
Information:					
9	Non Patent Literature	Mo_comparisonwithTCPReno.pdf	254976 e2694951a7c99413a789939b6e5c030c45ed1667	no	19
Warnings:					
Information:					

RED HAT 1012
RED HAT V. CAS

10	Non Patent Literature	Mo_Infocom1999.pdf	697459 487eb0e4fc295319b013ada2f3e23c620bb16d77	no	8
Warnings:					
Information:					
11	Non Patent Literature	Rodriguez_Biersack_IEEE_8-2002.pdf	1244725 3bf972b21b4751cc379ee237438d5163084504f0	no	11
Warnings:					
Information:					
12	Non Patent Literature	Srisuresh_RFC2663_8-1999.pdf	78195 a9ce72da9df53ca628b740158b9bf4f4a450c4f9	no	30
Warnings:					
Information:					
13	Non Patent Literature	Steinbach_iwdc_tcp_02.pdf	193150 90f367cda64e2f92f72cc3568d0a97652a262f92	no	8
Warnings:					
Information:					
14	Non Patent Literature	Balakrishnan_IEEE_12-1997_fonts.pdf	493337 e7f09c77a6ba4716c1d334f37e258b3b55cec227	no	17
Warnings:					
Information:					
15	Non Patent Literature	Chen_IEEE_Efficient_MultiPath_3-1998_fonts.pdf	433733 2d8331ba6ffccd2d368d38f4e397e02692a5ffcd	no	9
Warnings:					
Information:					
16	Non Patent Literature	Mascolo_ACMMobicom_2001_fonts.pdf	296126 3572fdc36f517dce01859938fef11dd983cf37c6	no	11
Warnings:					
Information:					
		Total Files Size (in bytes):	17024467		

RED HAT 1012
RED HAT V. CAS

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Mail Stop **Amendment**
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

AMENDMENT

In response to the Office Action issued on April 21, 2014 ("Office Action"), please reconsider the above-identified application in light of the following amendments to the claims and the accompanying remarks. This Amendment is being submitted on or before July 21, 2014 via EFS-Web.

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION

I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: July 21, 2014

By: Clay McGurk
Clay McGurk

RED HAT 1012
RED HAT V. CAS

THE CLAIMS

This listing of claims will replace all prior versions and listings of the claims in the application:

1-16. (Canceled).

17. (Currently Amended) A method for increasing bandwidth and data throughput comprising the steps of:

sending a multilink request from a first device to a second device to join a multilink connection ~~establishing a wireless connection between a first device and a second device~~, the first device being connected at a first location to a network ~~via a landline~~, the second device being connected at a second location to the network, the second location being geographically separate from the first location, the first device and the second device being configured for accessing the network; and

when the multilink request is accepted by the second device, ~~concurrently~~ receiving at the first device first data from the network at the first location and during the multilink connection, concurrently receiving second data over ~~the~~ a wireless connection from the second device, the second data being received at the second location from the network.

18. (Previously Presented) The method of claim 17, wherein the first location and the second location are physically isolated from each other.

19. (Previously Presented) The method of claim 17, further comprising the step of aggregating the second data received over the wireless connection and the first data received over the connection at the first location.

20. (Previously Presented) The method of claim 17, wherein a connection at the first location and a connection at the second location are network connection points.

21. (Previously Presented) The method of claim 17, wherein the first device and the second device are consumer networking devices.

22. (Currently Amended) The method of claim 17, further comprising the steps of[;]: sending another multilink request from the first device to a third device, ~~establishing a second wireless connection between the first device and a third device,~~ the third device being connected at a third location to the network, the third location being geographically different than the first location and the second location; and

when the another multilink request is accepted by the third deivce, concurrently receiving third data during the multilink connection over ~~the~~ a second wireless connection between the first device and the third device ~~with the receipt of the first data at the first location and the second data over the wireless connection~~.

23. (Previously Presented) The method of claim 22, further comprising the step of aggregating the first data, the second data and the third data.

24. (Canceled).

25. (Previously Presented) The method of claim 17, wherein the first data and the second data are received as data packets.

26. (Previously Presented) The method of claim 17, wherein the network is the Internet.

27. (Previously Presented) The method of claim 17, wherein the second device is a wireless device.

28. (Previously Presented) The method of claim 17, wherein the network is one or a combination of a cable network, a telecommunication network, an optical network and a communications network.

29. (Previously Presented) The method of claim 17, wherein the wireless connection is a cellular connection, a satellite connection, a Wi-Fi connection, a WiMax connection or a Bluetooth connection.

30-34. (Canceled).

35. (Currently Amended) A device comprising:

an interface that establishes a ~~landline~~ network connection at a first location to a first network and receives first data packets over the network connection;

a wireless interface that is configured to establish ~~establishes~~ a wireless connection to a remote communications device, the remote communications device being connected at a second location to a second network~~, the second location being geographically separate from the first location~~; and

a processor that sends a multilink request over the wireless interface to the remote communications device to join the multilink connection, receives from the interface the first data packets from the first network and when the multilink request is accepted by the remote communications device, concurrently receives ~~over~~ from the wireless interface second data packets from the second network, the second data packets being received at the second location from the second network.

36. (Previously Presented) The device of claim 35, wherein the device is one of a modem, cellular phone, a router, a computer, hotspot device and a tablet.

37. (Previously Presented) The device of claim 35, wherein the device is part of a home-based network system.

38. (Canceled).

39. (Currently Amended) A method for providing broadband communications services, comprising:

sending to a remote host an inquiry in whether the remote host is able to support a multilink connection;

receiving first data packets over a ~~landline~~ first connection to the Internet;

when the remote host indicates that the multilink connection is supported, receiving second data packets over ~~a wireless~~ a second connection, the first data packets being concurrently received with the second data packets during the multilink connection; and

wherein the first data packets create a first data stream and the second data packets create a second data stream.

40. (Currently Amended) The method of claim 39, wherein the first data packets are received at a home-based network system, and wherein the second packets are received over the ~~wireless~~ second connection at the home-based network system.

41. (Currently Amended) The method of claim 39, wherein the first data packets are received by a consumer-based network system, and wherein the second packets are received over the ~~wireless~~ second connection at the consumer-based network system.

42. (Currently Amended) The method of claim 39, further comprising:

during the multilink connection, receiving data packets over a plurality of wireless connections, the data packets being concurrently received with the first data packets; and

aggregating the first data packets with the data packets.

RED HAT 1012
RED HAT V. CAS

43. (Currently Amended) A method for increasing bandwidth and data throughput comprising the steps of:

sending a multilink request from a first device to a second device to join a multilink connection ~~establishing a wireless connection between a first device and a wireless device,~~ the first device being connected at a first location to a first network, the wireless device being connected at a second location to the first network or a second network, the second location being geographically separate from the first location; and

when the multilink request is accepted by the second device, ~~concurrently~~ receiving at the first device first data from the first network at the first location and during the multilink connection, concurrently receiving second data over ~~the~~ a wireless connection from the wireless device, the second data being received at the second location from the first network or the second network.

RED HAT 1012
RED HAT V. CAS

REMARKS

Status of the Claims

Claims 17-23, 25-29, 35-37 and 39-43 were pending in this application before this Amendment.

In this Amendment, claims 17, 22, 35 and 39-43 have been amended.

Thus, claims 17-23, 25-29, 35-37 and 39-43 are currently pending in this application pursuant to this Amendment.

35 U.S.C. § 103 Rejections

On pages 2-4 of the Office Action, claims 17-18, 20-22, 25-29, 35-37, 39-41 and 43 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent 7,426,316 to Vehvilainen ("Vehvilainen") in view of U.S. Application Publication 2009/0150924 to Xu ("Xu"). Vehvilainen and Xu, in combination or individually, do not teach all the limitations of claims 17, 35, 39 and 43, and each reference is non-analogous art.

Claim limitations are not described or suggested in Vehvilianen or Xu

In the Office Action, the "first device" was equated to computer 104 of Vehvilainen, and the "second device" was equated to image server 106. (Office Action, last five lines on p. 2). Vehvilainen shows in Fig. 1 that computer 104 and image server 106 are connected to network 103. Xu was used to illustrate in Fig. 1 that receiver 14 can receive electronic programming guides ("EPG") from both content provider 12 and network 18. (Xu, ¶ 23; Office Action, p. 3, lines 5-7). However, both Vehvilainen and Xu are deficient in neither teaches as required in amended claims 17, 35, 39 and 43 anything about a (1) "multilink request" that is sent by a first device to a second device; or (2) the required "wireless connection" between two devices, where (3) the two devices are connected to and concurrently receiving data from the network, and where (4) one device is transferring its data received from the network to the other device during a multilink connection.

Vehvilainen and Xu fail to describe anything about a "multilink request" that is sent by a first device to a second device as required by claims 17, 35, 39 and 43. "Multilink requests" and "multink connections" are absent from any of the teachings of Vehvilainen and Xu. Although

RED HAT 1012
RED HAT V. CAS

the receiver 14 in Xu may receive data from network 18 and content provider 12 at the same time, this is not a "multilink connection" for the reasons discussed below

Vehvilainen fails to describe or suggest the required wireless connection between two devices, where the two devices are connected to and concurrently receiving data from the network, and where one device is transferring its data received from the network to the other device during a multilink connection, as required in claims 17, 35 and 43. Vehvilainen does not describe a "wireless connection" between image server 106 and computer 104. The Office Action does not state with particularity where Vehvilainen describes this feature between the image server 106 and computer 104. Vehvilainen only shows that computer 104 and image server 106 are each connected to network 103, but Fig. 1 does not show any "wireless connection" required by the claims from computer 104 to image server 106.

In Vehvilainen, data is transferred in one direction between image server 106 and computer 104, and visa versa. Vehvilainen only shows the transfer of data between networked devices, nothing more. Vehvilainen is significantly missing any description of both image server 106 and computer 104 concurrently receiving data from network 103, and where data received at computer 104 or image server 106 is also being transferred to the other device over a wireless connection during a multilink connection.

Even if phone 102 in Vehvilainen was substituted for image server 106 (as was done in the previous office action), phone 102 and computer 104 are only communicating in a point-to-point connection over the network 103 or via a Bluetooth wireless connection. Vehvilainen would still fall short because it does not describe phone 102 and computer 104 both concurrently receiving data from network 103, and where data received at phone 102 or computer 104 is also transferred to the other device during a multilink connection. Vehvilainen only shows data going one way, from phone 102 to computer 104, or visa versa, over a single connection. What would be required by claims 17, 35 and 43 is that image server 106 communicates data concurrently to phone 102 and computer 104, and while this data is being received at phone 102, phone 102 also would be sending data to computer 104. This is clearly not shown or described in Vehvilainen in any manner.

Xu fails to cure the deficiences of Vehvilainen. Xu shows in Fig. 1 a single connection between receiver 14 and the network 18. Xu is missing a description or figure that shows two receivers 14, where the two receivers 14 are connected to network 18 and to each other via a

wireless connection. This is essential and required by claims 17, 35 and 43. Xu would need to show data being sent by EPG server 20 to network 18, where two receivers 14 are concurrently receiving data from network 18, but where one receiver 14 is also sending its data to the other receiver 14 over a wireless connection during a multilink connection. In other words, a second receiver 14 is missing from Xu that is required by the claims. Xu therefore fails to show the required devices, the required connections to the network, and the required wireless connection between devices as required in claims 17, 35, 39 and 43. Xu is not able to cure the deficiencies of Vehvilianen.

In summary, the combination of Vehvilainen and Xu falls short because both references do not show, describe, mention or suggest as required in amended claims 17, 35, 39 and 43 anything about a (1) "multilink request" that is sent by a first device to a second device; or (2) the required "wireless connection" between two devices, where (3) the two devices are connected to and concurrently receiving data from the network, and where (4) one device is transferring its data received from the network to the other device during a multilink connection.

Vehvilainen and Xu are each Non-analogous Art

"In order for a reference to be proper for use in an obviousness rejection under 35 U.S.C. §103, the reference must be analogous art to the claimed invention. *In re Bigio*, 381 F.3d 1320, 1325 (Fed. Cir. 2004); *see also,* MPEP 2141.01(a). A reference is analogous art to the claimed invention if: (1) the reference is from the same field of endeavor as the claimed invention (even if it addresses a different problem); or (2) the reference is reasonably pertinent to the problem faced by the inventor (even if it is not in the same field of endeavor as the claimed invention). *See* Bigio, 381 F.3d at 1325. Vehvilainen and Xu are each non-analogous art according to this two prong test.

Under the first prong of the test, Vehvilainen and Xu have to be from the same field or endeavor as the claimed invention. Neither reference is from the same field or endeavor as the claimed invention. The core teachings of Vehvilainen are methods for enhancing image quality by means of store image distortion information in an image server 106 that is then used to correct images taken from a specific imaging devices 102. (*See,* Vehvilainen, Abstract). This field or endeavor as described in Vehvilainen has nothing to do with the multilink connection of the claimed invention.

Xu describes that its invention relates to the use of electronic programming guides to make selections for video viewing. (Xu, ¶2 and Abstract). Xu teaches how to use separate file indexes accessed through two separate media connections to access desired content. (Xu, ¶ 22). The desired content (e.g., video) is only transmitted over the primary transmission medium (i.e., "A" in Fig. 1), not both transmission media. (Xu, ¶ 24). When content is being provided to receiver via route "A", receiver 14 may access more programming guide information via route "B". (Xu, ¶24). Providing programming guide information via two separate routes to a single receiver is not related to the same field or endeavor as the claimed invention. Neither Vehvilainen nor Xu are therefore from the same field or endeavor as the claimed invention, e.g., how to create a multilink connection, and thus fail the first prong of the test for analogous art.

Under the second prong of the test, Vehvilainen and Xu each has to be reasonably pertinent to the problem faced by the inventor of the present application. Again, the problems faced in Vehvilainen and Xu are not reasonably pertinent to the problem faced by the inventor of the present application. Vehvilainen is not trying to solve anything related to a multilink connection, but rather is using image correcting information to decrease errors in an image that is produced by a camera module. (Vehvilainen, Abstract). Similarly, Xu is trying to solve the problem related to downloading, accessing and using programming guides. (Xu, ¶¶ 2-5, 22, 24). Therefore, Vehvilainen and Xu fail the second prong of the test for analogous art.

In summary, Vehvilainen and Xu fail both prongs of the test for analogous art. Therefore, Vehvilainen and Xu are each non-analogous art and should not be used in an obviousness rejection for the claimed invention.

Summary

A combination of Vehvilainen and Xu does not result in the claimed invention because both fail to describe or suggest, either individually or in combination, each and all the elements required in claims 17, 35, 39 and 43. Vehvilainen and Xu are also non-analogous art and can not be used in an obviousness-type rejection. For these reasons, Vehvilainen in view of Xu lack the requirements for making a *prima facie* case of obviousness. Thus claims 17, 35, 39, and 43 are allowable over Vehvilainen in view of Xu. All the other claims that depend on these claims should be similarly allowed due to their dependence on those allowable independent claims. Therefore, it is respectfully requested that the rejection of claims 17-18, 20-22, 25-29, 35-37, 39-

41 and 43 under 35 U.S.C. §103(a) based on Vehvilainen and Xu be reconsidered and withdrawn.

Rejection of claims 19, 23 and 42

On page 4 of the Office Action, claims 19, 23 and 42 were rejected under 35 U.S.C. §103(a) as being unpatentable over Vehvilainen in view of Xu and U.S. Patent 7,653,089 to Jost *et al.* ("Jost"). The Office Action stated that Jost "discloses multiplexing data streams from various service providers (figure 7)". (Office Action, p. 4, § 4). This is being used against the "aggregating" limitation in claims 19, 23 and 42. Jost is not cited or relied upon for making up for the deficiencies of Vehvilainen and Xu as discussed above. Claims 19 and 23 are allowable due to their dependence on allowable independent claim 17. Claim 42 is similarly allowable due to its dependence on allowable independent claim 39. It is respectfully requested that the rejection under 35 U.S.C. §103 of claims 19, 23 and 42 as being unpatentable over Vehvilainen in view of Xu and Jost be reconsidered and withdrawn.

Conclusion

All rejections in the Office Action have been properly traversed. It is respectfully requested that all outstanding rejections be reconsidered and withdrawn. It is believed that a full and complete reply has been made in response to the Office Action, and, that as such, the present patent application is in a condition for allowance.

The Examiner is always welcome to telephone the attorney of record at the telephone number provided below if such telephone discussion would help expedite the prosecution of the patent application. Prompt and favorable consideration of this Amendment is respectfully requested.

Respectfully Submitted,

Date: July 21, 2014

H C McGurk IV, Reg. No. 34,964
Attorney for Applicant
The Law Office of Clay McGurk
PO Box 1488
Orange, CA 92706
Phone: **949.395.3942**
Email: claymcgurk@gmail.com

RED HAT 1012
RED HAT V. CAS

Electronic Acknowledgement Receipt

EFS ID:	19632352
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	21-JUL-2014
Filing Date:	22-JUN-2012
Time Stamp:	14:19:53
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Amendment/Req. Reconsideration-After Non-Final Reject	13531294_Amendment_7-21-2014.pdf	116417 2e87284a0ecc2be4ee1862958f9b8a4d9d6f4879	no	11

Warnings:	
Information:	

RED HAT 1012
RED HAT V. CAS

Total Files Size (in bytes):	116417

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

64828 7590 08/21/2014

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

EXAMINER	
CHO, HONG SOL	

ART UNIT	PAPER NUMBER
2467	

NOTIFICATION DATE	DELIVERY MODE
08/21/2014	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

CLAYMCGURK@GMAIL.COM

Office Action Summary	Application No. 13/531,294	Applicant(s) DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07/21/14*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5)☒ Claim(s) *17-23,25-29,35-37 and 39-43* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *17-23, 25-29, 35-37 and 39-43* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. This office action is in response to the amendment filed on 07/21/14. Claims 17-23, 25-29, 35-37 and 39-43 are pending in the instant application.

Claim Rejections - 35 USC § 112, First paragraph

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 17-23, 25-29, 35-37 and 39-43 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Re claims 17, 35 and 43, it recites "sending a multilink request from a first device to a second device to join a multilink connection". The original specification fails to describe the above claim limitation. Thus, Examiner concludes that the claimed feature indicated above is a new matter.

RED HAT 1012
RED HAT V. CAS

Re claim 39, it recites "sending to a remote host an inquiry in whether the remote host is able to support a multilink connection". The original specification fails to describe the above claim limitation. Thus, Examiner concludes that the claimed feature indicated above is a new matter.

Claims 18-23, 25-29, 36, 37 and 40-42 depend on claims 17, 35 and 39 are therefore similarly rejected.

Response to Arguments

4. Applicant's arguments with respect to claims have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

5. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the

RED HAT 1012
RED HAT V. CAS

advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hong Cho whose telephone number is 571-272-3087. The examiner can normally be reached on Mon-Fri during 7 am to 4 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hassan Phillips can be reached on 571-272-3940. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Hong Cho/

Primary Examiner, Art Unit 2467

RED HAT 1012
RED HAT V. CAS

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**		
	Application Number	13531294
	Filing Date	2012-06-22
	First Named Inventor	Eric Delangis
	Art Unit	2467
	Examiner Name	Hong Cho
	Attorney Docket Number	ED 1040

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.C./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	13531294 13531294 - GAU: 2467
	Filing Date	2012-06-22
	First Named Inventor	Eric Delangis
	Art Unit	2467
	Examiner Name	Hong Cho
	Attorney Docket Number	ED 1040

	1	H. Balakrishnan et al., "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", IEEE/ ACM Transactions on Networking, 17 pages; 1997.	☐
	2	J. Chen et al., "An Efficient Multipath Forwarding Method", Proceedings of IEEE Infocom '98; 9 pages; 1998.	☐
	3	J. Chen, "New Approaches to Routing for Large-Scale Data Networks", Thesis for Doctor of Philosophy, Rice University; 259 pages; 1999.	☐
	4	M. Handley et al., "Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics", Proceedings of the 10th USENIX Security Symposium; 18 pages; 2001.	☐
	5	H. Hsieh et al., "A Transport Layer Approach for Achieving Aggregate Bandwidths on Multi-homed Mobile Hosts", Proc. of MobiCom; 12 pages; August 2002.	☐
	6	E. Kohler et al., "The Click Modular Router", Proc. of the 17th Symposium on Operating Systems Principles; 34 pages; 1999.	☐
	7	S. Lee et al., "Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks", Infocom 98 - Proceedings, IEEE (Volume 3); 5 pages; 1998.	☐
	8	Y. Lee et al., "Improving TCP Performance in Multipath Packet Forwarding Networks", Journal of Communications and Networks, Vol. 4, No. 2; 10 pages; June 2002.	☐
	9	L. Magalhaes et al., "Transport Level Mechanisms for Bandwidth Aggregation on Mobile Hosts", IEEE-ICNP; 7 pages; 2001.	☐
	10	S. Mascolo et al., "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", Proc. of the ACM Mobicom; 11 pages; 2001.	☐
	11	J. Mo et al., "Analysis and Comparison of TCP Reno and Vegas", University of California at Berkeley; 19 pages; 1998.	☐

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.C./

RED HAT 1012
RED HAT V. CAS

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	13531294 13531294 - GAU: 2467
	Filing Date	2012-06-22
	First Named Inventor	Eric Delangis
	Art Unit	2467
	Examiner Name	Hong Cho
	Attorney Docket Number	ED 1040

	12	J. Mo et al., "Analysis and Comparison of TCP Reno and Vegas", IEEE; 8 pages; 1999.	☐
	13	P. Rodriguez et al., "Dynamic Parallel Access to Replicated Content in the Internet", IEEE/ACM Transactions on Networking; 11 pages; 2000.	☐
	14	P. Srisuresh et al., "Request for Comments: 2663 - IP Network Address Translator (NAT) Terminology and Considerations"; The Internet Society; 30 pages; 1999.	☐
	15	E. Steinbach et al., "A Simulation Study of Packet Path Diversity for TCP File Transfer and Media Transport on the Internet", Stanford University; 8 pages; August 2002.	☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/Hong Cho/	Date Considered	07/23/2014

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

EFS Web 2.1.17 ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.C./

RED HAT 1012
RED HAT V. CAS

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	13531294 13531294 - GAU: 2467
	Filing Date	2012-06-22
	First Named Inventor	Eric Delangis
	Art Unit	2467
	Examiner Name	Hong Cho
	Attorney Docket Number	ED 1040

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☒ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☐ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/H C McGurk IV/	Date (YYYY-MM-DD)	2014-06-03
Name/Print	H C McGurk	Registration Number	34964

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.C./

RED HAT 1012
RED HAT V. CAS

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.C./

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L2	32	multi near link with ppp same request$3	US-PGPUB; USPAT	OR	OFF	2014/08/18 10:50
L7	41	(request$3 or join$3 or establish$3) with multi near link with connection	US-PGPUB; USPAT	OR	OFF	2014/08/18 11:55
L8	10	(eric near delangis).in.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2014/08/18 12:38
L9	156	(multiplex$3 or aggregat$3 or combin$3) with wireless$2 near received near (stream$1 or flow$1 or traffic or data or message$1 or inforamtion or packet$1 or frame$1)	US-PGPUB; USPAT	OR	OFF	2014/08/18 12:39
L10	16	L9 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2014/08/18 12:39
L12	59094	(370/235,230,233,363,371,374,381,433,465-468,470-474,483-486,493-495,352,395,252).ccls.	US-PGPUB; USPAT	OR	OFF	2014/08/18 12:41
L16	58	L12 and multi near link with ppp	US-PGPUB; USPAT	OR	OFF	2014/08/18 12:58
L17	10	L16 and @rlad<"20021015"	US-PGPUB; USPAT	OR	OFF	2014/08/18 12:58
L18	7	(join$3 or request$3 or support$3) with multi near link near connection	US-PGPUB; USPAT	OR	OFF	2014/08/18 13:04

8/18/2014 1:05:17 PM
C:\Users\hcho\Documents\EAST\Workspaces\13531294.wsp

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	Art Unit
	HONG CHO	2467

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
370	230-235,278,282,328,338,468,477	11/26/2012	hc

SEARCH NOTES		
Search Notes	Date	Examiner
EAST text search	11/26/2012	hc
updated search	5/21/2013	hc
updated search	9/5/2013	hc
updated search	12/23/2013	hc
updated search	4/16/2014	hc
updated search	8/18/2014	hc

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	11/26/2012	05/21/2013	09/05/2013	12/23/2013	04/16/2014	08/18/2014			
	1	✓	✓	-	-	-	-			
	2	✓	✓	-	-	-	-			
	3	✓	✓	-	-	-	-			
	4	✓	✓	-	-	-	-			
	5	✓	✓	-	-	-	-			
	6	✓	✓	-	-	-	-			
	7	✓	✓	-	-	-	-			
	8	✓	✓	-	-	-	-			
	9		✓	-	-	-	-			
	10		✓	-	-	-	-			
	11		✓	-	-	-	-			
	12		✓	-	-	-	-			
	13		✓	-	-	-	-			
	14		✓	-	-	-	-			
	15		✓	-	-	-	-			
	16		✓	-	-	-	-			
	17			✓	✓	✓	✓			
	18			✓	✓	✓	✓			
	19			✓	✓	✓	✓			
	20			✓	✓	✓	✓			
	21			✓	✓	✓	✓			
	22			✓	✓	✓	✓			
	23			✓	✓	✓	✓			
	24			✓	-	-	-			
	25			✓	✓	✓	✓			
	26			✓	✓	✓	✓			
	27			✓	✓	✓	✓			
	28			✓	✓	✓	✓			
	29			✓	✓	✓	✓			
	30			✓	-	-	-			
	31			✓	-	-	-			
	32			✓	-	-	-			
	33			✓	-	-	-			
	34			✓	-	-	-			
	35			✓	✓	✓	✓			
	36			✓	✓	✓	✓			

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	11/26/2012	05/21/2013	09/05/2013	12/23/2013	04/16/2014	08/18/2014			
	37			✓	✓	✓	✓			
	38				✓	-	-			
	39				✓	✓	✓			
	40				✓	✓	✓			
	41				✓	✓	✓			
	42				✓	✓	✓			
	43					✓	✓			

To: CLAYMCGURK@GMAIL.COM,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

Aug 21, 2014 05:33:30 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	CTFR	08/21/2014	ED 1040
	1449	08/21/2014	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Mail Stop **AF**
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

REPLY UNDER 37 C.F.R. §1.116

In response to the Office Action issued on August 21, 2014 ("Office Action"), please reconsider the above-identified application in light of the following remarks. This Reply is being submitted before November 21, 2014 via EFS-Web.

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION
I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: August 25, 2014 By: Clay McGurk
Clay McGurk

MAILSTOP AF

RED HAT 1012
RED HAT V. CAS

THE CLAIMS

This listing of claims will replace all prior versions and listings of the claims in the application:

1-16. (Canceled).

17. (Previously Presented) A method for increasing bandwidth and data throughput comprising the steps of:

sending a multilink request from a first device to a second device to join a multilink connection, the first device being connected at a first location to a network, the second device being connected at a second location to the network, the second location being geographically separate from the first location, the first device and the second device being configured for accessing the network; and

when the multilink request is accepted by the second device, receiving at the first device first data from the network at the first location and during the multilink connection, concurrently receiving second data over a wireless connection from the second device, the second data being received at the second location from the network.

18. (Previously Presented) The method of claim 17, wherein the first location and the second location are physically isolated from each other.

19. (Previously Presented) The method of claim 17, further comprising the step of aggregating the second data received over the wireless connection and the first data received over the connection at the first location.

20. (Previously Presented) The method of claim 17, wherein a connection at the first location and a connection at the second location are network connection points.

21. (Previously Presented) The method of claim 17, wherein the first device and the second device are consumer networking devices.

22. (Previously Presented) The method of claim 17, further comprising the steps of: sending another multilink request from the first device to a third device, the third device being connected at a third location to the network, the third location being geographically different than the first location and the second location; and

when the another multilink request is accepted by the third device, concurrently receiving third data during the multilink connection over a second wireless connection between the first device and the third device.

23. (Previously Presented) The method of claim 22, further comprising the step of aggregating the first data, the second data and the third data.

24. (Canceled).

25. (Previously Presented) The method of claim 17, wherein the first data and the second data are received as data packets.

26. (Previously Presented) The method of claim 17, wherein the network is the Internet.

27. (Previously Presented) The method of claim 17, wherein the second device is a wireless device.

28. (Previously Presented) The method of claim 17, wherein the network is one or a combination of a cable network, a telecommunication network, an optical network and a communications network.

29. (Previously Presented) The method of claim 17, wherein the wireless connection is a cellular connection, a satellite connection, a Wi-Fi connection, a WiMax connection or a Bluetooth connection.

30-34. (Canceled).

35. (Previously Presented) A device comprising:

an interface that establishes a network connection at a first location to a first network and receives first data packets over the network connection;

a wireless interface that is configured to establish a wireless connection to a remote communications device, the remote communications device being connected at a second location to a second network; and

a processor that sends a multilink request over the wireless interface to the remote communications device to join the multilink connection, receives from the interface the first data packets from the first network and when the multilink request is accepted by the remote communications device, concurrently receives from the wireless interface second data packets from the second network, the second data packets being received at the second location from the second network.

36. (Previously Presented) The device of claim 35, wherein the device is one of a modem, cellular phone, a router, a computer, hotspot device and a tablet.

37. (Previously Presented) The device of claim 35, wherein the device is part of a home-based network system.

38. (Canceled).

39. (Previously Presented) A method for providing broadband communications services, comprising:

sending to a remote host an inquiry in whether the remote host is able to support a multilink connection;

receiving first data packets over a first connection to the Internet;

when the remote host indicates that the multilink connection is supported, receiving second data packets over a second connection, the first data packets being concurrently received with the second data packets during the multilink connection; and

wherein the first data packets create a first data stream and the second data packets create a second data stream.

40. (Previously Presented) The method of claim 39, wherein the first data packets are received at a home-based network system, and wherein the second packets are received over the second connection at the home-based network system.

41. (Previously Presented) The method of claim 39, wherein the first data packets are received by a consumer-based network system, and wherein the second packets are received over the second connection at the consumer-based network system.

42. (Previously Presented) The method of claim 39, further comprising:

during the multilink connection, receiving data packets over a plurality of wireless connections, the data packets being concurrently received with the first data packets; and

aggregating the first data packets with the data packets.

43. (Previously Presented) A method for increasing bandwidth and data throughput comprising the steps of:

sending a multilink request from a first device to a second device to join a multilink connection the first device being connected at a first location to a first network, the wireless device being connected at a second location to the first network or a second network, the second location being geographically separate from the first location; and

when the multilink request is accepted by the second device, receiving at the first device first data from the first network at the first location and during the multilink connection, concurrently receiving second data over a wireless connection from the wireless device, the second data being received at the second location from the first network or the second network.

RED HAT 1012
RED HAT V. CAS

REMARKS

Status of the Claims

Claims 17-23, 25-29, 35-37 and 39-43 are currently pending in this application. No amendments are being presented herein.

35 U.S.C. §112 Rejection

On pages 2-3 of the Office Action, claims 17-23, 25-29, 35-37 and 39-43 were rejected under 35 U.S.C. §112, first paragraph, as allegedly containing new matter for (1) "sending a multilink request from a first device to a second device to join a multilink connection" as used in claims 17, 35 and 43; and (2) "sending to a remote host an inquiry in whether the remote host is able to support a multilink connection" as used in claim 39. Both of these claim limitations are supported by the specification as detailed below.

In general, multilink connections are described over sixty times throughout the specification of the present application. Multilink connections are explained in great detail starting with paragraph 74 under the heading "Multilink PPP Connections" and continuing through paragraphs 79 of the specification of the present application. Fig. 7 of the present application illustrates multilink connections.

Support for "sending a multilink request from a first device to a second device to join a multilink connection" as used in claims 17, 35 and 43 can be found in paragraph 77 of the present application where it states (emphasis added):

> … The initiating RCG then selects which of the responding remote RCGs--86 it will use for the multilink PPP bundle--87 and sends them a multilink initialization packet containing the URL of the destination as well as the multilink PPP session ID (magic number) and any other pertinent information required for the remote RCG ***to join*** in the existing multilink PPP bundle--87…

Further support is found in paragraph 78 of the present application where it states (emphasis added):

> … Once a remote RCG--86 has been removed from participating in a multilink bundle--87, the initiating RCG--85 will then look up another remote RCG in its router tables that can be used to participate in the multilink PPP bundle--87, and request that the new remote RCG--86 ***to join*** the existing multilink bundle--87…

RED HAT 1012
RED HAT V. CAS

Therefore, there is clear support in the specification of the present application for "sending a multilink request from a first device to a second device to join a multilink connection" as used in claims 17, 35 and 43.

Support for "sending a remote host an inquiry in whether the remote host is able to support a multilink connection" as used in claim 39 can be found at the beginning of paragraph 77 of the present application where it states:

> Referring to FIG. 7, an RCG--85 that decides to initiate a multilink PPP session with a remote computer i.e. internet URL, requests a multilink PPP session with that URL resource via Link Control Protocol (LCP) session management. The initiating RCG--85 and the remote server computer (not shown) will negotiate the pertinent multilink parameters such as compression, packet size, sync/async, link quality monitoring, bit stuffing etc. …

Link Control Protocol (LCP) as referenced in paragraph 77 of the specification of the present application is an important protocol. To those of ordinary skill in the art, it is well known that Link Control Protocol has three primary functions: (1) link configuration; (2) link maintenance; and (3) link termination. The primary concern is with link configuration, where LCP link configuration is the process of setting up and negotiating the parameters of a link. In the link establishment phase of LCP link configuration, two devices exchange information about the conditions under which the link will operate. LCP link configuration is described in more detail in Appendix A, attached hereto, from "The TCP/IP Guide" (available on the Internet). This article describes LCP link configuration as known by those of ordinary skill in the art, and provides a flowchart illustrating the different steps of this process. This flowchart includes the steps of determining whether a multilink connection can be supported. If a multilink connection is possible, "Responding Device Sends Configure-Ack Message"; otherwise, "Responder Sends Configure-Reject Message". Therefore, there is clear support for the step of "sending a remote host an inquiry in whether the remote host is able to support a multilink connection" as in claim 39 from the specification of the present application itself, and additionally from the functionality of LCP as understood by those of ordinary skill in the art.

For these reasons, it is respectfully submitted that there is clear support in the specification of the present application for the steps founds in claims 17, 35, 39 and 43. It is therefore respectfully requested that the rejections of claims 17-23, 25-29, 35-37 and 39-43

under 35 U.S.C. §112, first paragraph be reconsidered and withdrawn, and this application be forthrightly allowed.

Conclusion

All rejections in the Office Action have been properly traversed. It is respectfully requested that all outstanding rejections be reconsidered and withdrawn. It is believed that a full and complete reply has been made in response to the Office Action, and, that as such, the present patent application is in a condition for allowance.

The Examiner is always welcome to telephone the attorney of record at the telephone number provided below if such telephone discussion would help expedite the prosecution of the patent application. Prompt and favorable consideration of this Reply is respectfully requested.

Respectfully Submitted,

Date: August 25, 2014

H C McGurk IV, Reg. No. 34,964
Attorney for Applicant

The Law Office of Clay McGurk
PO Box 1488
Orange, CA 92706

Phone: **949.395.3942**
Email: claymcgurk@gmail.com

RED HAT 1012
697
RED HAT V. CAS

APPENDIX A

RED HAT 1012
RED HAT V. CAS

The TCP/IP Guide A TCP/IP Reference You Can Understand!



NOTE: Using software to mass-download the site **degrades the server and is prohibited.** If you want to read The TCP/IP Guide offline, please consider licensing it. Thank you.



The Book is Here... and *Now On Sale!*

Contents

The TCP/IP Guide
9 TCP/IP Lower-Layer (Interface, Internet and Transport) Protocols (OSI Layers 2, 3 and 4)
9 TCP/IP Network Interface Layer (OSI Data Link Layer) Protocols
9 TCP/IP Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP)
9 Point-to-Point Protocol (PPP)
9 PPP Core Protocols: Link Control, Network Control and Authentication

Get The TCP/IP Guide for your own computer. The TCP/IP Guide

Prev Topic PPP Core Protocols: Link Control, Network Control and Authentication

Prev Page Pages 1 2 3 Next Page

PPP Network Control Protocols (IPCP, IPXCP, NBFCP and others) Next Topic

Search
Google™ Custom Search

▶ PPP
▶ TCP IP Protocol
▶ Protocol

PPP Link Control Protocol (LCP)

LCP Link Configuration

Link configuration is arguably the most important job that LCP does in PPP. During the *Link Establishment* phase, LCP frames are exchanged that enable the two physically-connected devices to negotiate the conditions under which the link will operate. Figure 27 shows the entire procedure, which we will now examine in detail.

The process starts with the initiating device (let's call it device *A*, yeah, isn't that original) creating a *Configure-Request* frame that contains a variable number of configuration options that it wants to see set up on the link. This is basically device *A*'s "wish list" for how it wants the link created.

699

RED HAT 1012
RED HAT V. CAS

I want to build a digital gorilla gram that I can send to my friends.

Launch on IBM Bluemix

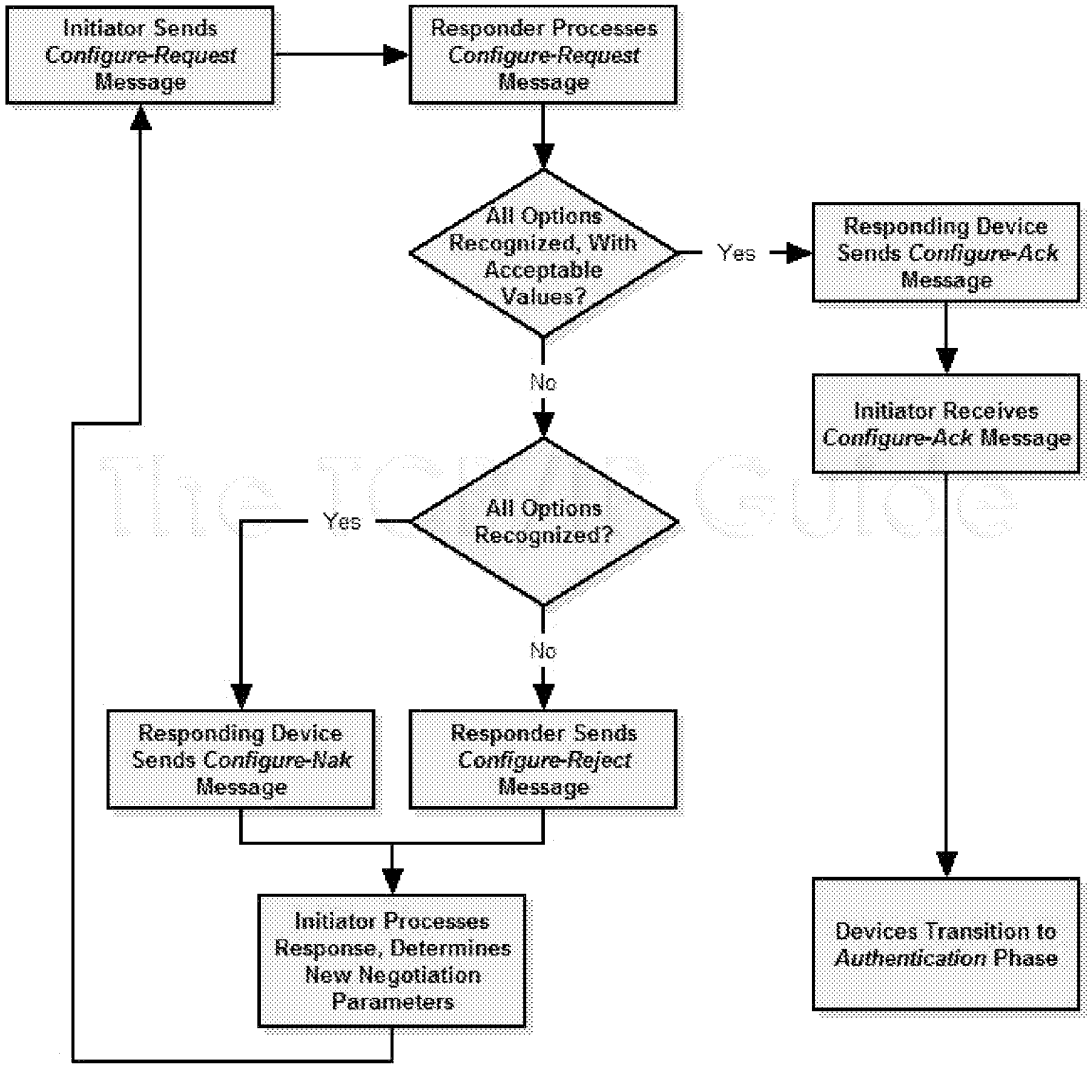


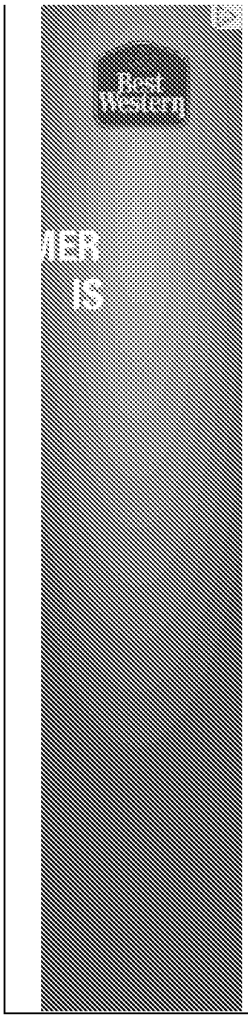
Figure 27: PPP LCP Link Configuration Process

This flowchart shows in more detail the negotiation process undertaken to configure the link by LCP. This process begins when the PPP link enters the *Link Establishment* phase. After successful configuration, the connection transitions to the *Authentication* phase.

The main PPP document (RFC 1661) defines a number of different configuration options that the initiator can specify in this request. Any one of these can be included and if so, filled in with the value corresponding to what device *A* wants for that option. If absent, this means device *A* is neither requesting nor specifying that option. The six options are:

- ***Maximum-Receive-Unit (MRU):*** Lets device *A* specify the maximum size datagram it wants the link to be able to carry.
- ***Authentication-Protocol:*** Device *A* can indicate the type of authentication protocol it wishes to use (if any).
- ***Quality-Protocol:*** If device *A* wants to enable quality monitoring on the link, what protocol to use (though there is only one currently defined: LQR).
- ***Magic-Number:*** Used to detect looped back links or other anomalies in the connection.

RED HAT 1012
RED HAT V. CAS



- ***Protocol-Field-Compression:*** Allows device *A* to specify that it wants to use "compressed" (8 bit) *Protocol* fields in PPP data frames instead of the normal 16 bit *Protocol* field. This provides a small but free savings (one byte) on each PPP frame. Note that this has nothing to do with the compression feature offered by CCP. See the PPP general frame format topic for more on this feature.

- ***Address-and-Control-Field-Compression (ACFC):*** The same as the option just above but used to compress the *Address* and *Control* fields, again for small bandwidth savings. Again, see the PPP general frame format topic for more.

Other options may also be added to this list by optional feature protocols. For example, Multilink PPP adds several options that must be negotiated during link setup.

The other device (let's call it say... device *B* ☺) receives the *Configure-Request* and processes it. It then has three choices of how to respond:

1. If every option in it is acceptable in every way, device *B* sends back a *Configure-Ack* ("acknowledge"). The negotiation is complete.

2. If all the options that device *A* sent are valid ones that device *B* recognizes and is capable of negotiating, but it doesn't accept the values device *A* sent, then device *B* returns a *Configure-Nak* ("negative acknowledge") frame. This message includes a copy of each configuration option that *B* found unacceptable.

3. If any of the options that *A* sent were either unrecognized by *B*, or represent ways of using the link that *B* considers not only unacceptable but not even subject to negotiation, it returns a *Configure-Reject* containing each of the objectionable options.

The difference between a *Configure-Nak* and a *Configure-Reject* is that the former is like device *B* saying "I don't accept your terms, but I'm willing to haggle", while the latter is device *B* basically saying "No way Jose". For example, if device *A* tries to request PAP as the authentication protocol but device *B* wants to use CHAP, it will send a *Configure-Nak*. If device *B* doesn't support authentication at all, it will send a *Configure-Reject*.



Note: Even after receiving a reject, device *A* can retry the negotiation with a new *Configure-Request*.

Prev Topic PPP Core Protocols: Link Control, Network Control and Authentication

Prev Page Pages 1 2 3 Next Page

PPP Network Control Protocols (IPCP, IPXCP, NBFCP and others) Next Topic

If you find The TCP/IP Guide useful, please consider making a small Paypal donation to help the site, using one of the buttons below. You can also donate a custom amount using the far right button (not less than $1 please, or PayPal gets most/all of your money!) In lieu of a larger donation, you may wish to consider purchasing a download license of The TCP/IP Guide. Thanks for your support!

Donate $2 Donate $5 Donate $10 Donate $20 Donate $30 Donate: $

PayPal DONATE

Home - Table Of Contents - Contact Us

The TCP/IP Guide (http://www.TCPIPGuide.com)
Version 3.0 - Version Date: September 20, 2005

© Copyright 2001-2005 Charles M. Kozierok. All Rights Reserved.
Not responsible for any loss resulting from the use of this site.

Electronic Acknowledgement Receipt

EFS ID:	19960863
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	25-AUG-2014
Filing Date:	22-JUN-2012
Time Stamp:	19:45:45
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Response After Final Action	13531294_Reply_8-25-2014.pdf	2838458 d2d5cb11d3a85ff0291386a75d8b2b6f4bce80b1	no	14

Warnings:	
Information:	

RED HAT 1012
RED HAT V. CAS

Total Files Size (in bytes):	2838458

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/531,294	Filing Date 06/22/2012	☐ To be Mailed

ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT **08/25/2014**

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 20	Minus	** 21	= 0	X $40 =	0
Independent (37 CFR 1.16(h))	* 4	Minus	***4	= 0	X $210 =	0
☐ Application Size Fee (37 CFR 1.16(s))						
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
					TOTAL ADD'L FEE	0

AMENDMENT

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))						
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
					TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/DONNA 1. SMALLS LOGAN/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

64828 7590 09/08/2014

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

EXAMINER
CHO, HONG SOL

ART UNIT	PAPER NUMBER
2467	

DATE MAILED: 09/08/2014

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

TITLE OF INVENTION: BROADBAND COMMUNICATIONS DEVICE

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0	$0	$480	12/08/2014

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 02/11)

RED HAT 1012
RED HAT V. CAS

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

64828 7590 09/08/2014
Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

TITLE OF INVENTION: BROADBAND COMMUNICATIONS DEVICE

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0	$0	$480	12/08/2014

EXAMINER	ART UNIT	CLASS-SUBCLASS
CHO, HONG SOL	2467	370-235000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:
- ❑ Issue Fee
- ❑ Publication Fee (No small entity discount permitted)
- ❑ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ❑ A check is enclosed.
- ❑ Payment by credit card. Form PTO-2038 is attached.
- ❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ❑ Applicant certifying micro entity status. See 37 CFR 1.29
  NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.
- ❑ Applicant asserting small entity status. See 37 CFR 1.27
  NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.
- ❑ Applicant changing to regular undiscounted fee status.
  NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

PTOL-85 Part B (10-13) Approved for use through 10/31/2013. OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

64828 7590 09/08/2014

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

EXAMINER	
CHO, HONG SOL	

ART UNIT	PAPER NUMBER
2467	

DATE MAILED: 09/08/2014

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

PTOL-85 (Rev. 02/11)

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

RED HAT 1012
RED HAT V. CAS

Notice of Allowability	Application No. 13/531,294	Applicant(s) DELANGIS, ERIC M.	
	Examiner HONG CHO	Art Unit 2467	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *08/25/14*.
   ☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on_____.
2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
3. ☒ The allowed claim(s) is/are *17-23,25-29,35-37,39-43*. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov .
4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   Certified copies:

   a) ☐ All b) ☐ Some *c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
4. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
5. ☐ Examiner's Amendment/Comment
6. ☐ Examiner's Statement of Reasons for Allowance
7. ☐ Other _____.

/HONG CHO/
Primary Examiner, Art Unit 2467

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

CPC						
Symbol					**Type**	**Version**
H04W		88	/	16	F	2013-01-01
H04L		12	/	5692	I	2013-01-01
H04W		8	/	26	A	2013-01-01
H04W		28	/	06	A	2013-01-01
H04W		40	/	02	A	2013-01-01
H04W		72	/	04	A	2013-01-01
H04W		80	/	00	A	2013-01-01
H04W		84	/	12	A	2013-01-01
H04W		84	/	18	A	2013-01-01
			/			
			/			
			/			
			/			
			/			
			/			

CPC Combination Sets							
Symbol					**Type**	**Set**	**Ranking**	**Version**
			/					
			/					

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	20	
/HONG CHO/ Primary Examiner.Art Unit 2467	09/02/14	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	7

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

US ORIGINAL CLASSIFICATION		INTERNATIONAL CLASSIFICATION										
CLASS	SUBCLASS	CLAIMED					NON-CLAIMED					
370	230	H	0	4	L	12 / 26 (2006.01.01)						
CROSS REFERENCE(S)		H	0	4	J	3 / 16 (2006.01.01)						
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)											
370	468											

NONE		Total Claims Allowed:	
(Assistant Examiner)	(Date)	20	
/HONG CHO/ Primary Examiner.Art Unit 2467	09/02/14	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	7

U.S. Patent and Trademark Office — Part of Paper No. 20140902

RED HAT 1012
RED HAT V. CAS

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	**Art Unit**
	HONG CHO	2467

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	17	17	40												
2	18	18	41												
3	19	19	42												
4	20	20	43												
5	21														
6	22														
7	23														
8	25														
9	26														
10	27														
11	28														
12	29														
13	35														
14	36														
15	37														
16	39														

NONE (Assistant Examiner)	(Date)	**Total Claims Allowed:** 20	
/HONG CHO/ Primary Examiner.Art Unit 2467 (Primary Examiner)	09/02/14 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 7

U.S. Patent and Trademark Office Part of Paper No. 20140902

RED HAT 1012
RED HAT V. CAS

REPLY UNDER 37 C.F.R. §1.116
EXPEDITED PROCEDURE
Technology Center 2400

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.:	**13/531,294**	EXAMINER:	Hong CHO
INVENTOR:	Eric M. Delangis	GROUP ART UNIT:	2467
FILED:	JUNE 22, 2012	CONFIRMATION NO.:	9530
TITLE:	A BROADBAND COMMUNICATIONS DEVICE		
ATTORNEY NO:	ED 1040		

Mail Stop **AF**
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

OK TO ENTER: /H.C./

REPLY UNDER 37 C.F.R. §1.116

In response to the Office Action issued on August 21, 2014 ("Office Action"), please reconsider the above-identified application in light of the following remarks. This Reply is being submitted before November 21, 2014 via EFS-Web.

CERTIFICATE OF ELECTRONIC (EFS-WEB) TRANSMISSION
I hereby certify that this correspondence is being transmitted via the U.S. Patent Office's electronic filing system in accordance with 37 C.F.R. § 1.8(a)(1) from the **Pacific Time Zone** of the United States on the local date shown below.

Dated: August 25, 2014 By: Clay McGurk
Clay McGurk

MAILSTOP AF

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 9530

SERIAL NUMBER	FILING or 371(c) DATE RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
13/531,294	06/22/2012	370	2467	ED 1040

APPLICANTS

INVENTORS

Eric M. Delangis, Huntsville, AL;

**** CONTINUING DATA ***************************

This application is a CON of 12/581,852 10/19/2009 PAT 8228801
which is a CON of 10/686,375 10/14/2003 PAT 7606156 *
which claims benefit of 60/418,521 10/15/2002
(*)Data provided by applicant is not consistent with PTO records.

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED **** ** SMALL ENTITY **
07/03/2012

Foreign Priority claimed	☐ Yes ☑ No	☑ Met after Allowance	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
35 USC 119(a-d) conditions met	☐ Yes ☑ No					
Verified and Acknowledged	/HONG SOL CHO/ Examiner's Signature	hc Initials	AL	9	8	3

ADDRESS

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

TITLE

BROADBAND COMMUNICATIONS DEVICE

FILING FEE RECEIVED		
740	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

BIB (Rev. 05/07).

RED HAT 1012
RED HAT V. CAS

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	59504	(370/235,230,233,363,371,374,381,433,465-468,470-474,483-486,493-495,352,395,252).ccls.	US-PGPUB; USPAT	OR	OFF	2014/09/02 10:01
L2	58	L1 and multi near link with ppp	US-PGPUB; USPAT	OR	OFF	2014/09/02 10:01
L5	10	(eric near delangis).in.	US-PGPUB; USPAT	OR	OFF	2014/09/02 10:11
L6	9	(join$3 or request$3 or support$3) with multi near link near connection	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2014/09/02 10:11
L7	11	((request$3 or join$3 or establish$3) with multi near link with connection).clm.	US-PGPUB; USPAT	OR	OFF	2014/09/02 10:44
L8	20	((multiplex$3 or aggregat$3 or combin$3) with wireless$2 near received near (stream$1 or flow$1 or traffic or data or message$1 or inforamtion or packet$1 or frame$1)).clm.	US-PGPUB; USPAT	OR	OFF	2014/09/02 11:09

9/2/2014 11:11:43 AM
C:\Users\hcho\Documents\EAST\Workspaces\13531294.wsp

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13531294	DELANGIS, ERIC M.
	Examiner	Art Unit
	HONG CHO	2467

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
370	230-235,278,282,328,338,468,477	11/26/2012	hc

SEARCH NOTES		
Search Notes	Date	Examiner
EAST text search	11/26/2012	hc
updated search	5/21/2013	hc
updated search	9/5/2013	hc
updated search	12/23/2013	hc
updated search	4/16/2014	hc
updated search	8/18/2014	hc
updated search	9/2/2014	hc

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner
370	230, 468	9/2/2014	hc

To: CLAYMCGURK@GMAIL.COM,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

Sep 08, 2014 05:28:55 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	NOA	09/08/2014	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

64828 7590 09/08/2014
Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	06/22/2012	Eric M. Delangis	ED 1040	9530

TITLE OF INVENTION: BROADBAND COMMUNICATIONS DEVICE

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0	$0	$480	12/08/2014

EXAMINER	ART UNIT	CLASS-SUBCLASS
CHO, HONG SOL	2467	370-235000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 **CLAY MCGURK**
2 **The Law Office of Clay McGurk**
3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

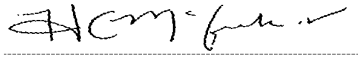
Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
☒ Issue Fee
☐ Publication Fee (No small entity discount permitted)
☐ Advance Order - # of Copies .....

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
☐ A check is enclosed.
☐ Payment by credit card. Form PTO-2038 is attached.
☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
☐ Applicant certifying micro entity status. See 37 CFR 1.29
NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.
☐ Applicant asserting small entity status. See 37 CFR 1.27
NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.
☐ Applicant changing to regular undiscounted fee status.
NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature  Date **September 12, 2014**

Typed or printed name **H C McGurk IV** Registration No. **34,964**

PTOL-85 Part B (10-13) Approved for use through 10/31/2013. OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

RED HAT 1012
RED HAT V. CAS

Electronic Patent Application Fee Transmittal

Application Number:	13531294
Filing Date:	22-Jun-2012
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Filer:	Harold C. McGurk
Attorney Docket Number:	ED 1040

Filed as Small Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl Issue Fee	2501	1	480	480
Publ. Fee- Early, Voluntary, or Normal	1504	1	0	0

RED HAT 1012
RED HAT V. CAS

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**480**

RED HAT 1012
RED HAT V. CAS

Electronic Acknowledgement Receipt

EFS ID:	20126175
Application Number:	13531294
International Application Number:	
Confirmation Number:	9530
Title of Invention:	BROADBAND COMMUNICATIONS DEVICE
First Named Inventor/Applicant Name:	Eric M. Delangis
Customer Number:	64828
Filer:	Harold C. McGurk
Filer Authorized By:	
Attorney Docket Number:	ED 1040
Receipt Date:	12-SEP-2014
Filing Date:	22-JUN-2012
Time Stamp:	15:39:09
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$480
RAM confirmation Number	1984
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)

RED HAT 1012
RED HAT V. CAS

1	Issue Fee Payment (PTO-85B)	13531294_PartB_9-12-2014.pdf	3103487 d1118cb735568dbb2b27c2916a0451e77beb0bd6	no	1
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	31865 b0e2ae4c7ba81583e066277343ba0359493156c5	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	3135352	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RED HAT 1012
RED HAT V. CAS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/531,294	10/14/2014	8861349	ED 1040	9530

64828 7590 09/24/2014

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 0 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Eric M. Delangis, Huntsville, AL;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

To: CLAYMCGURK@GMAIL.COM,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 64828

Sep 25, 2014 05:22:49 AM

Dear PAIR Customer:

Harold McGurk
The Law Office of Clay McGurk
P.O. BOX 1488
Orange, CA 92856
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 64828 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
13531294	ISSUE.NTF	09/24/2014	ED 1040

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

RED HAT 1012
RED HAT V. CAS