

FOURTH EDITION

Computer Networks

ANDREW S. TANENBAUM



Computer Networks

Fourth Edition

Andrew S. Tanenbaum

*Vrije Universiteit
Amsterdam, The Netherlands*



Prentice Hall PTR
Upper Saddle River, NJ 07458
www.phptr.com

Library of Congress Cataloging-in-Publication Data

Tanenbaum, Andrew S.
Computer networks / Andrew S. Tanenbaum.--4th ed.
p. cm.
Includes bibliographical references.
ISBN 0-13-066102-3
1. Computer networks. I. Title.
TK5105.5 .T36 2002
004.6--dc21

2002029263

Editorial/production supervision: *Patti Guerrieri*
Cover design director: *Jerry Votta*
Cover designer: *Anthony Gemmellaro*
Cover design: *Andrew S. Tanenbaum*
Art director: *Gail Cocker-Bogusz*
Interior Design: *Andrew S. Tanenbaum*
Interior graphics: *Hadel Studio*
Typesetting: *Andrew S. Tanenbaum*
Manufacturing buyer: *Maura Zaldivar*
Executive editor: *Mary Franz*
Editorial assistant: *Noreen Regina*
Marketing manager: *Dan DePasquale*



© 2003, 1996 Pearson Education, Inc.
Publishing as Prentice Hall PTR
Upper Saddle River, New Jersey 07458

Prentice Hall books are widely used by corporations and government agencies for training, marketing, and resale.

For information regarding corporate and government bulk discounts please contact:
Corporate and Government Sales (800) 382-3419 or corpsales@pearsontechgroup.com

All products or services mentioned in this book are the trademarks or service marks of their respective companies or organizations.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2

ISBN 0-13-066102-3

Pearson Education LTD.
Pearson Education Australia PTY, Limited
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd.
Pearson Education Canada, Ltd.
Pearson Educación de México, S.A. de C.V.
Pearson Education — Japan
Pearson Education Malaysia, Pte. Ltd.

Fifth, it must be possible to start out with one or two devices and expand the reach of the network gradually. This means no format wars. Telling consumers to buy peripherals with IEEE 1394 (FireWire) interfaces and a few years later retracting that and saying USB 2.0 is the interface-of-the-month is going to make consumers skittish. The network interface will have to remain stable for many years; the wiring (if any) will have to remain stable for decades.

Sixth, security and reliability will be very important. Losing a few files to an e-mail virus is one thing; having a burglar disarm your security system from his PDA and then plunder your house is something quite different.

An interesting question is whether home networks will be wired or wireless. Most homes already have six networks installed: electricity, telephone, cable television, water, gas, and sewer. Adding a seventh one during construction is not difficult, but retrofitting existing houses is expensive. Cost favors wireless networking, but security favors wired networking. The problem with wireless is that the radio waves they use are quite good at going through fences. Not everyone is overjoyed at the thought of having the neighbors piggybacking on their Internet connection and reading their e-mail on its way to the printer. In Chap. 8 we will study how encryption can be used to provide security, but in the context of a home network, security has to be foolproof, even with inexperienced users. This is easier said than done, even with highly sophisticated users.

In short, home networking offers many opportunities and challenges. Most of them relate to the need to be easy to manage, dependable, and secure, especially in the hands of nontechnical users, while at the same time delivering high performance at low cost.

1.2.6 Internetworks

Many networks exist in the world, often with different hardware and software. People connected to one network often want to communicate with people attached to a different one. The fulfillment of this desire requires that different, and frequently incompatible networks, be connected, sometimes by means of machines called **gateways** to make the connection and provide the necessary translation, both in terms of hardware and software. A collection of interconnected networks is called an **internetwork** or **internet**. These terms will be used in a generic sense, in contrast to the worldwide Internet (which is one specific internet), which we will always capitalize.

A common form of internet is a collection of LANs connected by a WAN. In fact, if we were to replace the label "subnet" in Fig. 1-9 by "WAN," nothing else in the figure would have to change. The only real technical distinction between a subnet and a WAN in this case is whether hosts are present. If the system within the gray area contains only routers, it is a subnet; if it contains both routers and hosts, it is a WAN. The real differences relate to ownership and use.

An interesting question is how a router turns a flow specification into a set of specific resource reservations. That mapping is implementation specific and is not standardized. Suppose that a router can process 100,000 packets/sec. If it is offered a flow of 1 MB/sec with minimum and maximum packet sizes of 512 bytes, the router can calculate that it might get 2048 packets/sec from that flow. In that case, it must reserve 2% of its CPU for that flow, preferably more to avoid long queueing delays. If a router's policy is never to allocate more than 50% of its CPU (which implies a factor of two delay, and it is already 49% full, then this flow must be rejected. Similar calculations are needed for the other resources.

The tighter the flow specification, the more useful it is to the routers. If a flow specification states that it needs a *Token bucket rate* of 5 MB/sec but packets can vary from 50 bytes to 1500 bytes, then the packet rate will vary from about 3500 packets/sec to 105,000 packets/sec. The router may panic at the latter number and reject the flow, whereas with a minimum packet size of 1000 bytes, the 5 MB/sec flow might have been accepted.

Proportional Routing

Most routing algorithms try to find the best path for each destination and send all traffic to that destination over the best path. A different approach that has been proposed to provide a higher quality of service is to split the traffic for each destination over multiple paths. Since routers generally do not have a complete overview of network-wide traffic, the only feasible way to split traffic over multiple routes is to use locally-available information. A simple method is to divide the traffic equally or in proportion to the capacity of the outgoing links. However, more sophisticated algorithms are also available (Nelakuditi and Zhang, 2002).

Packet Scheduling

If a router is handling multiple flows, there is a danger that one flow will hog too much of its capacity and starve all the other flows. Processing packets in the order of their arrival means that an aggressive sender can capture most of the capacity of the routers its packets traverse, reducing the quality of service for others. To thwart such attempts, various packet scheduling algorithms have been devised (Bhatti and Crowcroft, 2000).

One of the first ones was the **fair queueing** algorithm (Nagle, 1987). The essence of the algorithm is that routers have separate queues for each output line, one for each flow. When a line becomes idle, the router scans the queues round robin, taking the first packet on the next queue. In this way, with n hosts competing for a given output line, each host gets to send one out of every n packets. Sending more packets will not improve this fraction.

Although a start, the algorithm has a problem: it gives more bandwidth to hosts that use large packets than to hosts that use small packets. Demers et al.

(1990) suggested an improvement in which the round robin is done in such a way as to simulate a byte-by-byte round robin, instead of a packet-by-packet round robin. In effect, it scans the queues repeatedly, byte-for-byte, until it finds the tick on which each packet will be finished. The packets are then sorted in order of their finishing and sent in that order. The algorithm is illustrated in Fig. 5-36.

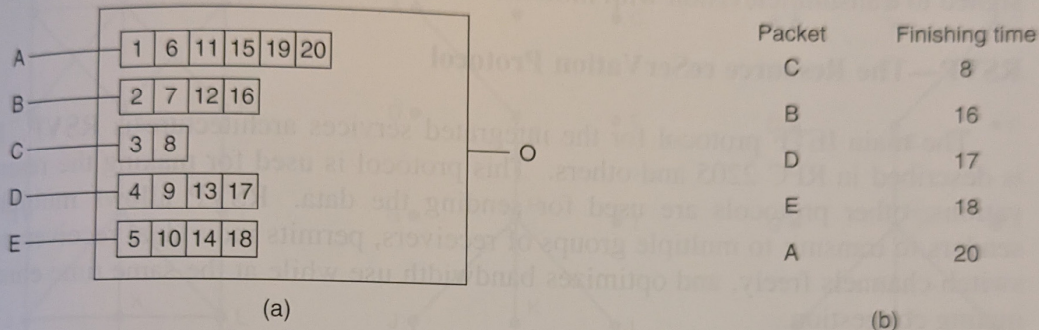


Figure 5-36. (a) A router with five packets queued for line O. (b) Finishing times for the five packets.

In Fig. 5-36(a) we see packets of length 2 to 6 bytes. At (virtual) clock tick 1, the first byte of the packet on line A is sent. Then goes the first byte of the packet on line B, and so on. The first packet to finish is C, after eight ticks. The sorted order is given in Fig. 5-36(b). In the absence of new arrivals, the packets will be sent in the order listed, from C to A.

One problem with this algorithm is that it gives all hosts the same priority. In many situations, it is desirable to give video servers more bandwidth than regular file servers so that they can be given two or more bytes per tick. This modified algorithm is called **weighted fair queueing** and is widely used. Sometimes the weight is equal to the number of flows coming out of a machine, so each process gets equal bandwidth. An efficient implementation of the algorithm is discussed in (Shreedhar and Varghese, 1995). Increasingly, the actual forwarding of packets through a router or switch is being done in hardware (Elhanany et al., 2001).

5.4.3 Integrated Services

Between 1995 and 1997, IETF put a lot of effort into devising an architecture for streaming multimedia. This work resulted in over two dozen RFCs, starting with RFCs 2205–2210. The generic name for this work is **flow-based algorithms** or **integrated services**. It was aimed at both unicast and multicast applications. An example of the former is a single user streaming a video clip from a news site. An example of the latter is a collection of digital television stations broadcasting their programs as streams of IP packets to many receivers at various locations. Below we will concentrate on multicast, since unicast is a special case of multicast.

such as a leaky bucket with some specified drain rate. An operator with a nose for business might charge extra for each premium packet transported or might allow up to N premium packets per month for a fixed additional monthly fee. Note that this scheme requires no advance setup, no resource reservation, and no time-consuming end-to-end negotiation for each flow, as with integrated services. This makes DS relatively easy to implement.

Class-based service also occurs in other industries. For example, package delivery companies often offer overnight, two-day, and three-day service. Airlines offer first class, business class, and cattle class service. Long-distance trains often have multiple service classes. Even the Paris subway has two service classes. For packets, the classes may differ in terms of delay, jitter, and probability of being discarded in the event of congestion, among other possibilities (but probably not roomier Ethernet frames).

To make the difference between flow-based quality of service and class-based quality of service clearer, consider an example: Internet telephony. With a flow-based scheme, each telephone call gets its own resources and guarantees. With a class-based scheme, all the telephone calls together get the resources reserved for the class telephony. These resources cannot be taken away by packets from the file transfer class or other classes, but no telephone call gets any private resources reserved for it alone.

Expedited Forwarding

The choice of service classes is up to each operator, but since packets are often forwarded between subnets run by different operators, IETF is working on defining network-independent service classes. The simplest class is **expedited forwarding**, so let us start with that one. It is described in RFC 3246.

The idea behind expedited forwarding is very simple. Two classes of service are available: regular and expedited. The vast majority of the traffic is expected to be regular, but a small fraction of the packets are expedited. The expedited packets should be able to transit the subnet as though no other packets were present. A symbolic representation of this "two-tube" system is given in Fig. 5-39. Note that there is still just one physical line. The two logical pipes shown in the figure represent a way to reserve bandwidth, not a second physical line.

One way to implement this strategy is to program the routers to have two output queues for each outgoing line, one for expedited packets and one for regular packets. When a packet arrives, it is queued accordingly. Packet scheduling should use something like weighted fair queueing. For example, if 10% of the traffic is expedited and 90% is regular, 20% of the bandwidth could be dedicated to expedited traffic and the rest to regular traffic. Doing so would give the expedited traffic twice as much bandwidth as it needs in order to provide low delay for it. This allocation can be achieved by transmitting one expedited packet for every four regular packets (assuming the size distribution for both classes is

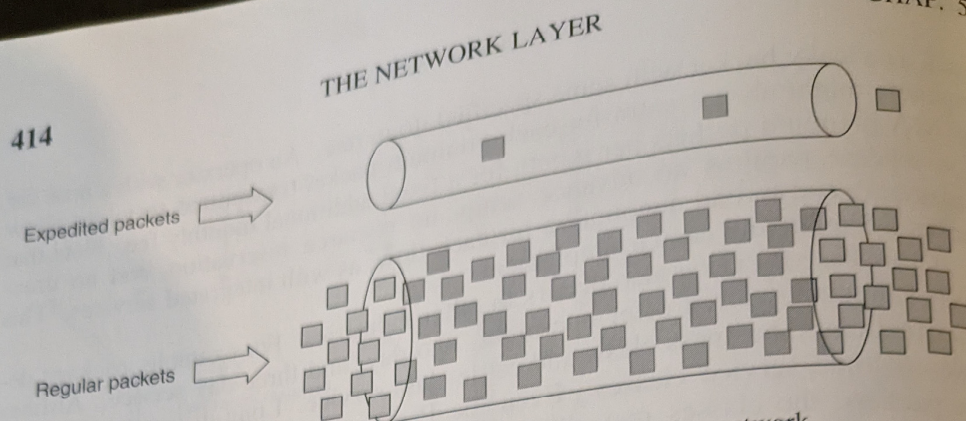


Figure 5-39. Expedited packets experience a traffic-free network, similar). In this way, it is hoped that expedited packets see an unloaded subnet, even when there is, in fact, a heavy load.

Assured Forwarding

A somewhat more elaborate scheme for managing the service classes is called **assured forwarding**. It is described in RFC 2597. It specifies that there shall be four priority classes, each class having its own resources. In addition, it defines three discard probabilities for packets that are undergoing congestion: low, medium, and high. Taken together, these two factors define 12 service classes.

Figure 5-40 shows one way packets might be processed under assured forwarding. Step 1 is to classify the packets into one of the four priority classes. This step might be done on the sending host (as shown in the figure) or in the ingress (first) router. The advantage of doing classification on the sending host is that more information is available about which packets belong to which flows there.

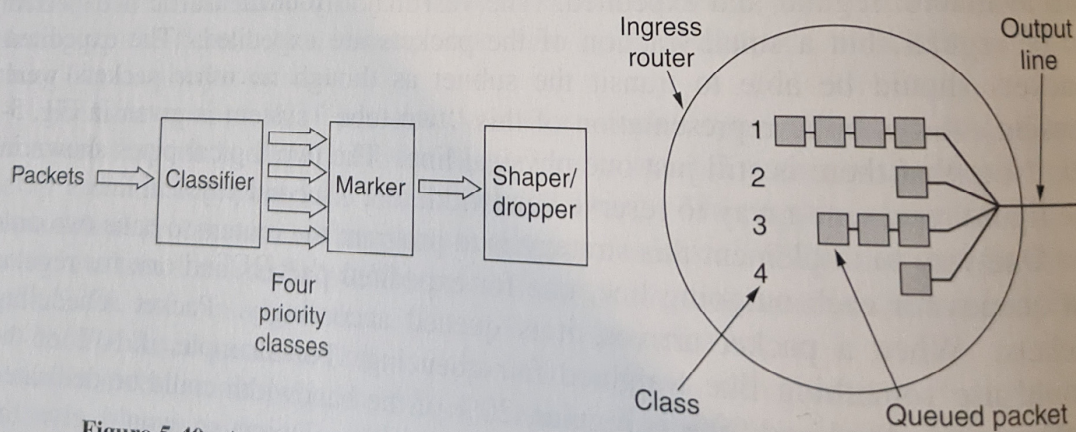


Figure 5-40. A possible implementation of the data flow for assured forwarding.

Step 2 is to mark the packets according to their class. A header field is needed for this purpose. Fortunately, an 8-bit *Type of service* field is available in the