8477231

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

April 10, 2024

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: ***17/142,983***
FILING DATE: ***January 06, 2021***
PATENT NUMBER: ***11695823***
ISSUE DATE: ***July 04, 2023***

Certified by

Performing the Functions and Duties of the
Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82A (07-13)
Approved for use through 03/31/2021. 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.

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	
Filing Date	
First Named Inventor	Taaghol
Title	Distributed Software Defined Network
Art Unit	
Examiner Name	
Attorney Docket Number	TAAG001CONT3

SIGNATURE of Applicant or Patent Practitioner

Signature	/Christopher J. Horgan/	Date (Optional)	January 6, 2021
Name	Christopher J. Horgan	Registration Number	40394
Title (if Applicant is a juristic entity)			
Applicant Name (if Applicant is a juristic entity)	Big Data Federation, Inc.		

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ *Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 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.

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 03/31/2021. 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 BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date
15/836,824	12/9/2017

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 112449

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

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

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

Big Data Federation, Inc.

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature		Date (Optional)	1/26/2020
Name	Pouya Taaghol		
Title	CEO, Big Data Federation, Inc.		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 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.

PTO/AIA/96 (08-12)
Approved for use through 11/30/2020. 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 displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

Applicant/Patent Owner: Big Data Federation, Inc.

Application No./Patent No.: _____ Filed/Issue Date: _____

Titled: Distributed Software Defined Networking

Big Data Federation, Inc., a corporation

(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that, for the patent application/patent identified above, it is (choose **one** of options 1, 2, 3 or 4 below):

1. ☑ The assignee of the entire right, title, and interest.

2. ☐ An assignee of less than the entire right, title, and interest (check applicable box):

   ☐ The extent (by percentage) of its ownership interest is _____%. Additional Statement(s) by the owners holding the balance of the interest must be submitted to account for 100% of the ownership interest.

   ☐ There are unspecified percentages of ownership. The other parties, including inventors, who together own the entire right, title and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

3. ☐ The assignee of an undivided interest in the entirety (a complete assignment from one of the joint inventors was made). The other parties, including inventors, who together own the entire right, title, and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

4. ☐ The recipient, via a court proceeding or the like (*e.g.*, bankruptcy, probate), of an undivided interest in the entirety (a complete transfer of ownership interest was made). The certified document(s) showing the transfer is attached.

The interest identified in option 1, 2 or 3 above (not option 4) is evidenced by either (choose **one** of options A or B below):

A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

B. ☑ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Pouya Taaghol To: Big Data Federation, Inc.

   The document was recorded in the United States Patent and Trademark Office at
   Reel 050963, Frame 0054, or for which a copy thereof is attached.

2. From: Vivek Ramanna To: Pouya Taaghol

   The document was recorded in the United States Patent and Trademark Office at
   Reel 035569, Frame 0334, or for which a copy thereof is attached.

This collection of information is required by 37 CFR 3.73(b). 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. 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.

PTO/AIA/96 (08-12)
Approved for use through 11/30/2020. 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 displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

3. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

4. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

5. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

6. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).

☑ As required by 37 CFR 3.73(c)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

/Christopher J. Horgan/	January 6, 2021
Signature	Date
Christopher J. Horgan	40394
Printed or Typed Name	Title or Registration Number

PTO/SB/35 (07-09)
Approved for use through 11/30/2020. 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 displays a valid OMB control number.

NONPUBLICATION REQUEST UNDER 35 U.S.C. 122(b)(2)(B)(i)

First Named Inventor	Taaghol
Title	Distributed Software Defined Networking
Attorney Docket Number	TAAG001CONT3

I hereby 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.

I hereby request that the attached application not be published under 35 U.S.C. 122(b).

/Christopher J. Horgan/ — Signature

January 6, 2020 — Date

Christopher J. Horgan — Typed or printed name

40493 — Registration Number, if applicable

408-687-8306 — Telephone Number

This request must be signed in compliance with 37 CFR 1.33(b) and submitted with the application **upon filing.**

Applicant may rescind this nonpublication request at any time. If applicant rescinds a request that an application not be published under 35 U.S.C. 122(b), the application will be scheduled for publication at eighteen months from the earliest claimed filing date for which a benefit is claimed.

If applicant subsequently files an application directed to the invention disclosed in the attached application in another country, or under a multilateral international agreement, that requires publication of applications eighteen months after filing, the applicant **must** notify the United States Patent and Trademark Office of such filing within forty-five (45) days after the date of the filing of such foreign or international application. **Failure to do so will result in abandonment of this application (35 U.S.C. 122(b)(2)(B)(iii)).**

This collection of information is required by 37 CFR 1.213(a). 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 6 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 (1-800-786-9199) and select option 2.

Doc Code: ECOMM.AUTH/ECOMM.WTDW
Doc Description: Internet Communications Authorized/Internet Communications Authorization Withdrawn

PTO/SB/439 (11-15)

AUTHORIZATION FOR INTERNET COMMUNICATIONS IN A PATENT APPLICATION OR REQUEST TO WITHDRAW AUTHORIZATION FOR INTERNET COMMUNICATIONS		
	Application No.	
	Filing Date	
	First Named Inventor	Taaghol
	Art Unit	
	Examiner Name	
	Practitioner Docket No.	TAAG001CONT3

To: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. To authorize permission for Internet Communications.

☑ Recognizing that Internet communications are not secure, I hereby authorize the USPTO to communicate with the undersigned and practitioners in accordance with 37 CFR 1.33 and 37 CFR 1.34 concerning any subject matter of this application via video conferencing, instant messaging, or electronic mail. I understand that a copy of these communications will be made of record in the application file. (MPEP 502.03)

II. To withdraw authorization for Internet Communications.

☐ The authorization given on _____, to the USPTO to communicate with the undersigned and any practitioner in accordance with 37 CFR 1.33 and 37 CFR 1.34 concerning any subject matter of this application via Internet communications such as video conferencing, instant messaging, or electronic mail is hereby withdrawn. I understand that the withdrawal is effective when approved rather than when received.

I am the

☐ applicant.

☑ attorney or agent of record. Registration number 40394.

☐ attorney or agent acting under 37 CFR 1.34. Registration number _____.

/Christopher J. Horgan/ — Signature

January 6, 2021 — Date

Christopher J. Horgan — Typed or printed name

408-687-8306 — Telephone Number

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Juristic entities must be represented by a patent practitioner (see 37 CFR 1.31, which is applicable to any paper filed on or after September 16, 2012, that is presented on behalf of a juristic entity, regardless of application filing date). Submit multiple forms if more than one signature is required, see below*.

☑ * Total of 1 forms are submitted.

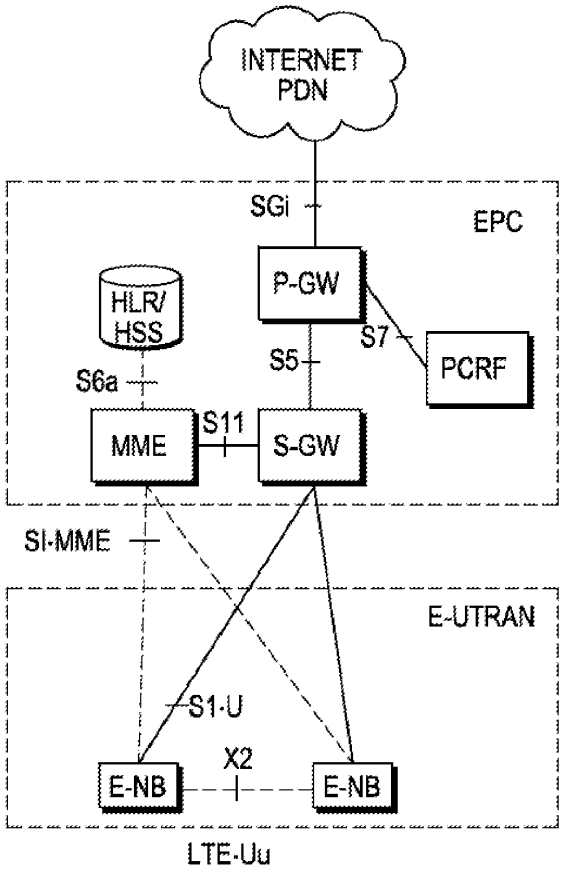

INTERNET
PDN
SGi
EPC
P-GW
HLR/
HSS
S7
PCRF
S5
S6a
S11
MME
S-GW
SI-MME
E-UTRAN
S1-U
X2
E-NB
E-NB
LTE-Uu
100

FIG. 1

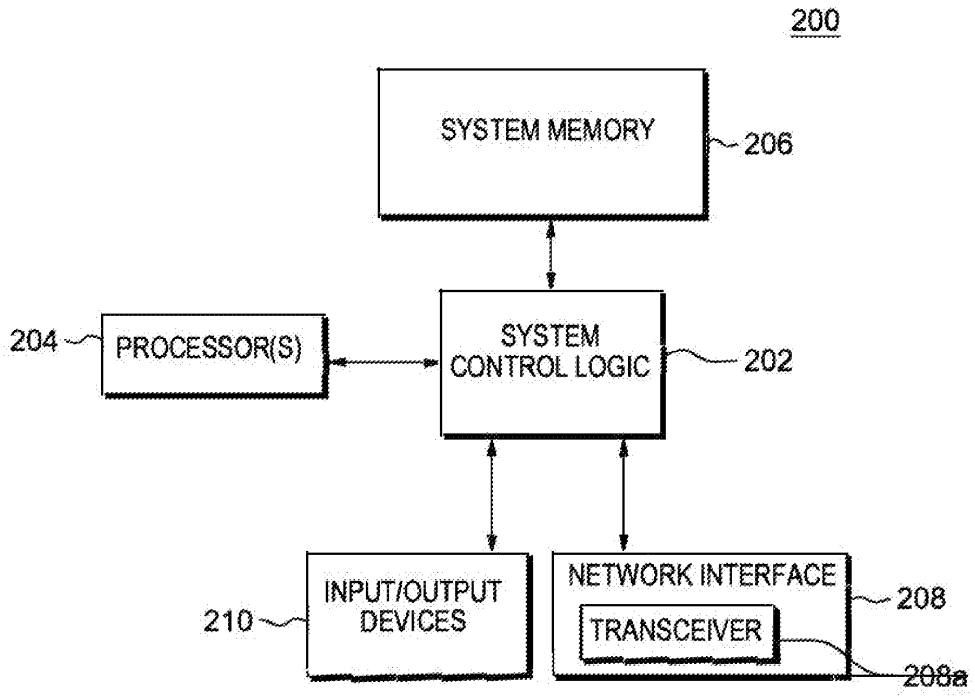

200
SYSTEM MEMORY
206
204
PROCESSOR(S)
SYSTEM
CONTROL LOGIC
202
210
INPUT/OUTPUT
DEVICES
NETWORK INTERFACE
208
TRANSCEIVER
208a

FIG. 2

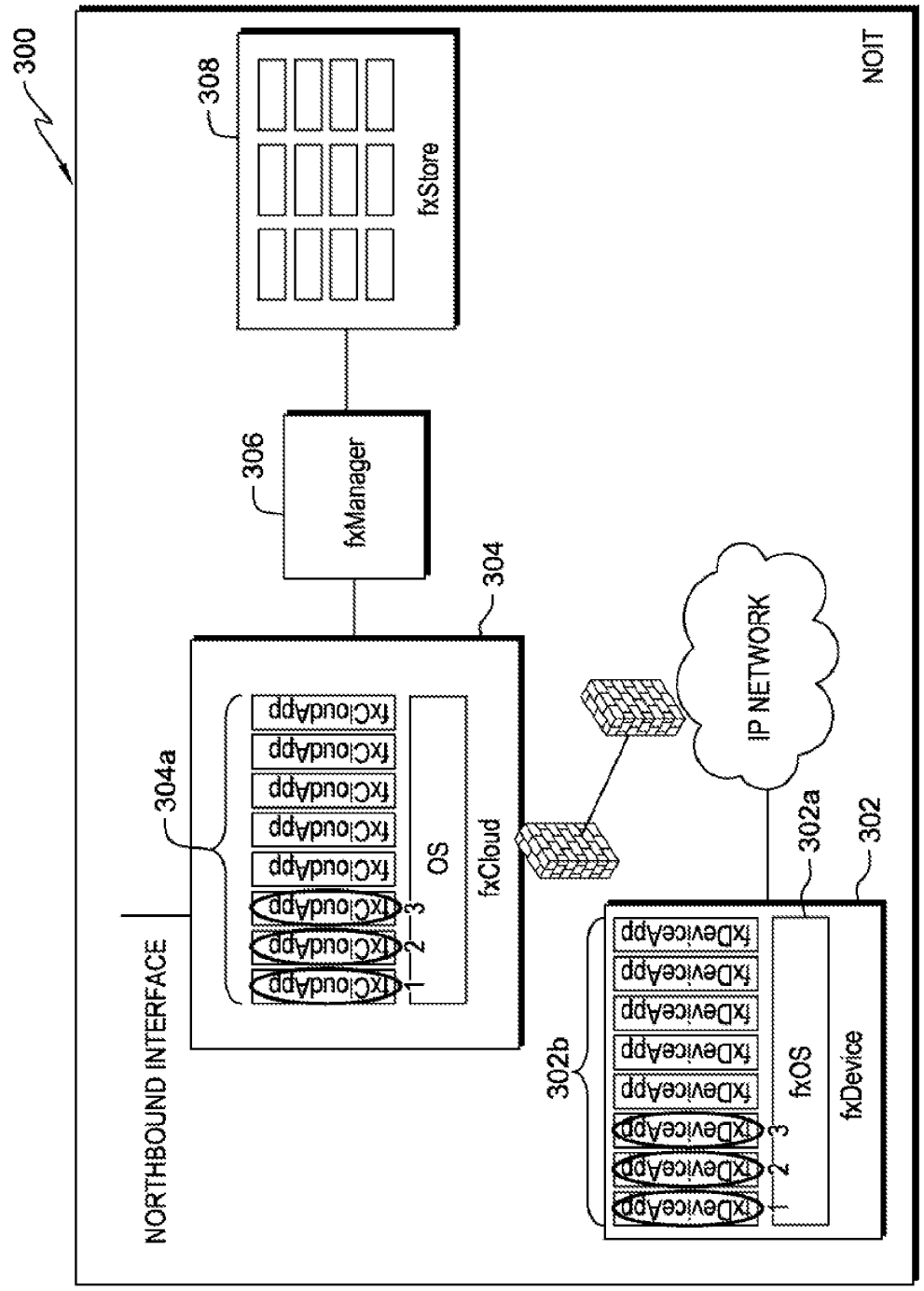

300
NOIT
308
fxStore
306
fxManager
304
304a
fxCloudApp
OS
fxCloud
IP NETWORK
NORTHBOUND INTERFACE
302a
302
302b
fxDeviceApp
fxOS
fxDevice
1
2
3

FIG. 3

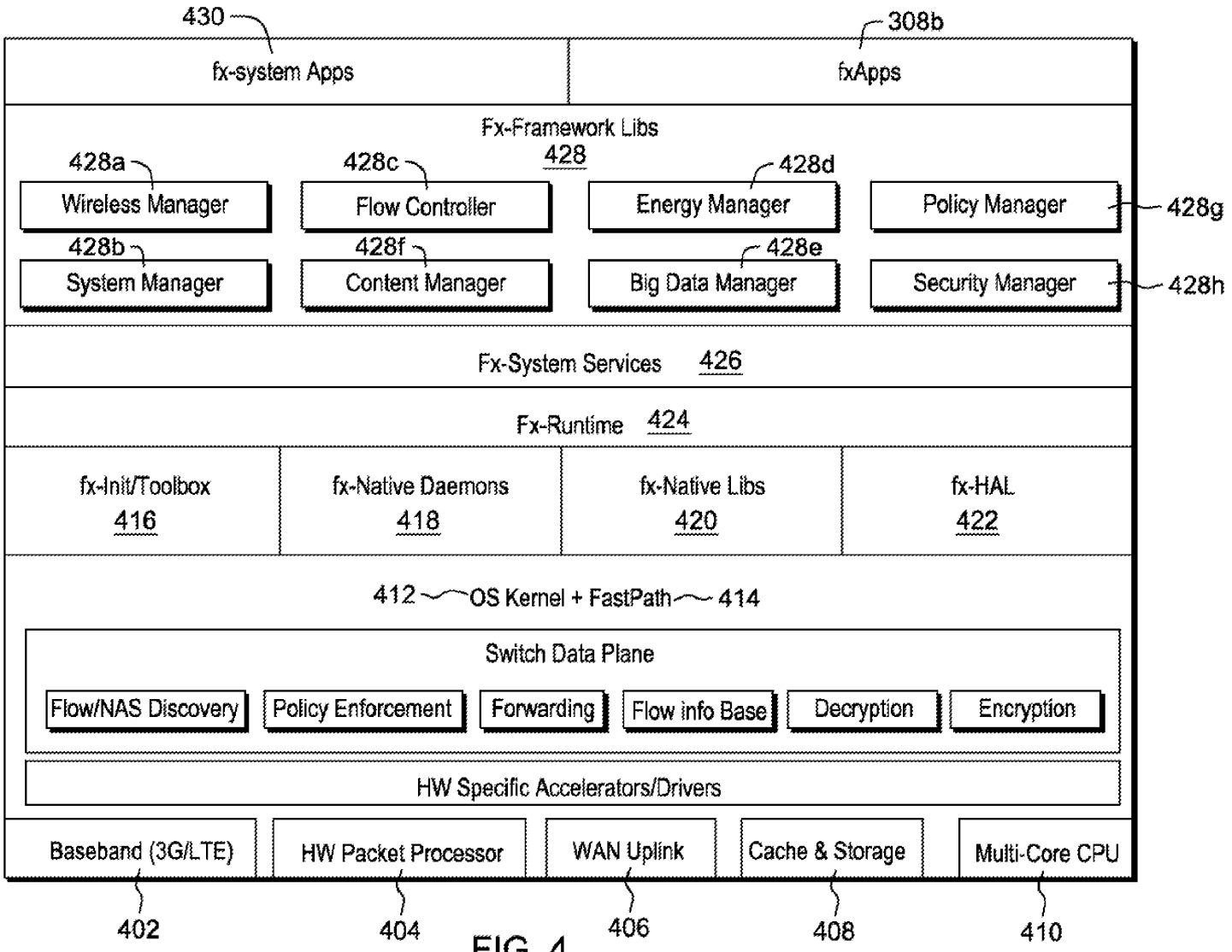

430
fx-system Apps
308b
fxApps
Fx-Framework Libs
428
428a
Wireless Manager
428c
Flow Controller
428d
Energy Manager
Policy Manager
428g
428b
System Manager
428f
Content Manager
428e
Big Data Manager
Security Manager
428h
Fx-System Services 426
Fx-Runtime 424
fx-Init/Toolbox
416
fx-Native Daemons
418
fx-Native Libs
420
fx-HAL
422
412
OS Kernel + FastPath
414
Switch Data Plane
Flow/NAS Discovery
Policy Enforcement
Forwarding
Flow info Base
Decryption
Encryption
HW Specific Accelerators/Drivers
Baseband (3G/LTE)
HW Packet Processor
WAN Uplink
Cache & Storage
Multi-Core CPU
402
404
406
408
410

FIG. 4

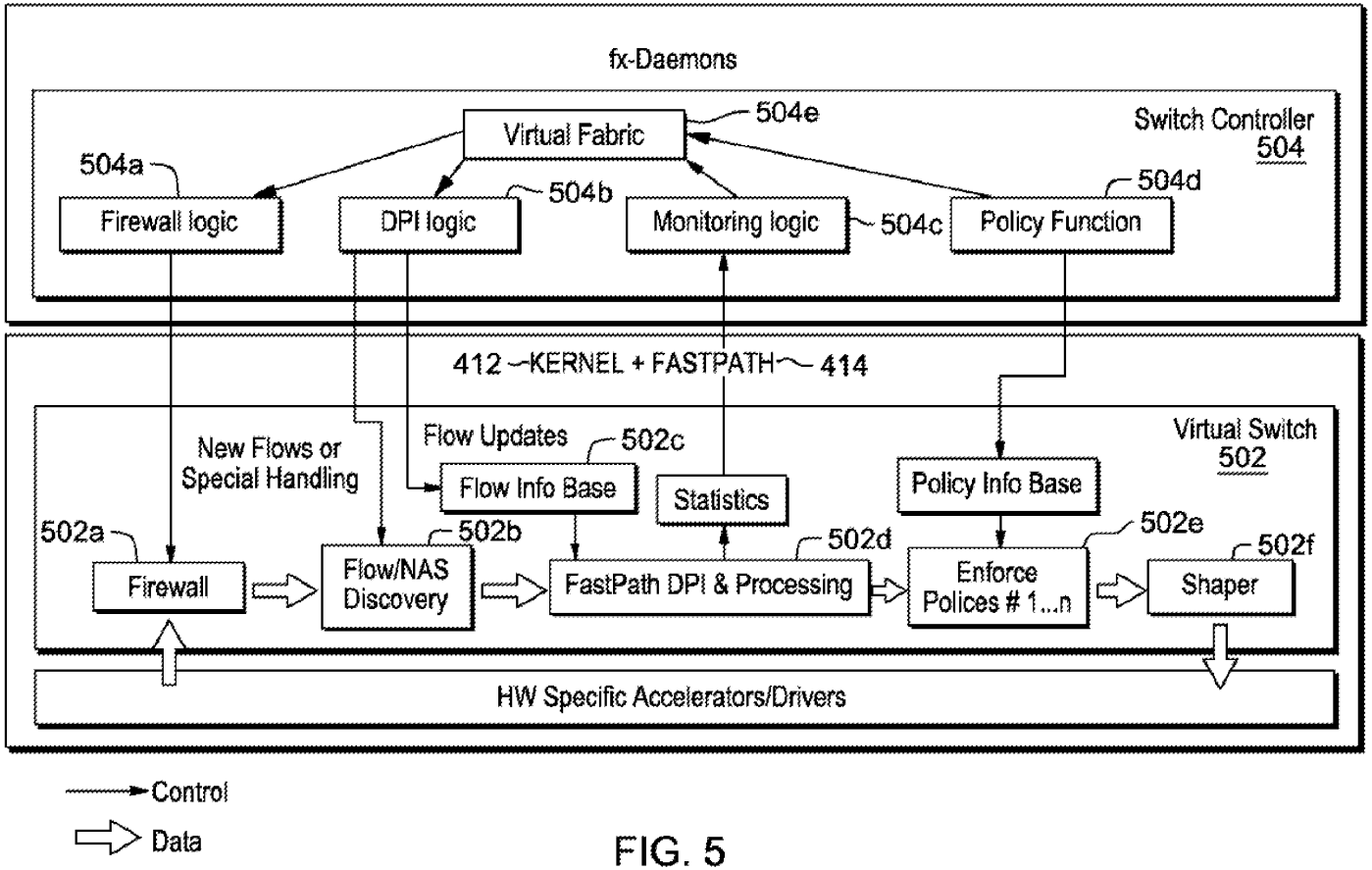

fx-Daemons
Virtual Fabric
504e
Switch Controller
504
504a
Firewall logic
DPI logic
504b
Monitoring logic
504c
Policy Function
504d
412
KERNEL + FASTPATH
414
New Flows or
Special Handling
Flow Updates
502c
Flow Info Base
Statistics
Policy Info Base
Virtual Switch
502
502a
Firewall
Flow/NAS
Discovery
502b
FastPath DPI & Processing
502d
Enforce
Polices # 1...n
502e
Shaper
502f
HW Specific Accelerators/Drivers
Control
Data

FIG. 5

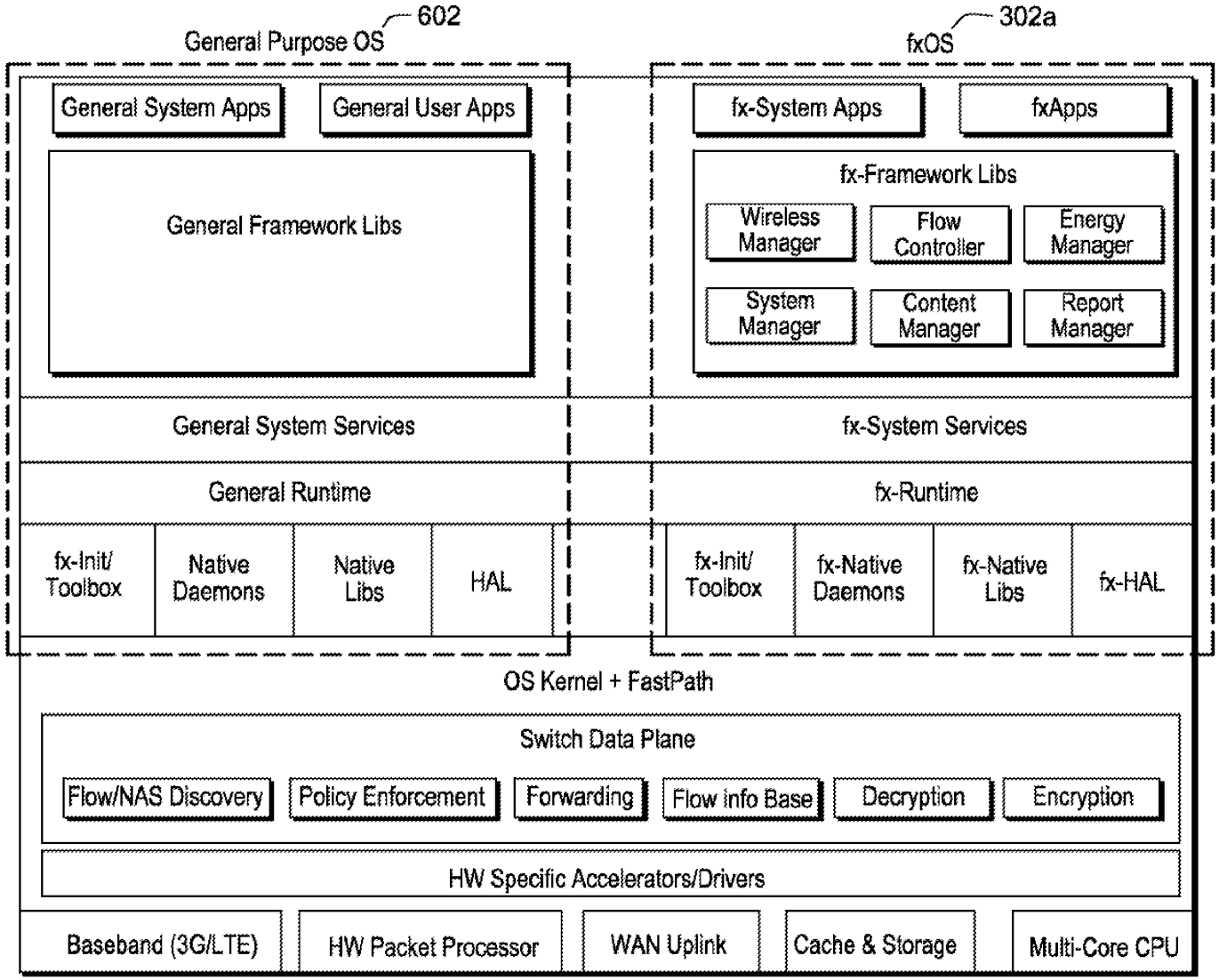

602
General Purpose OS
302a
fxOS
General System Apps
General User Apps
General Framework Libs
fx-System Apps
fxApps
fx-Framework Libs
Wireless Manager
Flow Controller
Energy Manager
System Manager
Content Manager
Report Manager
General System Services
fx-System Services
General Runtime
fx-Runtime
fx-Init/ Toolbox
Native Daemons
Native Libs
HAL
fx-Init/ Toolbox
fx-Native Daemons
fx-Native Libs
fx-HAL
OS Kernel + FastPath
Switch Data Plane
Flow/NAS Discovery
Policy Enforcement
Forwarding
Flow info Base
Decryption
Encryption
HW Specific Accelerators/Drivers
Baseband (3G/LTE)
HW Packet Processor
WAN Uplink
Cache & Storage
Multi-Core CPU

FIG. 6

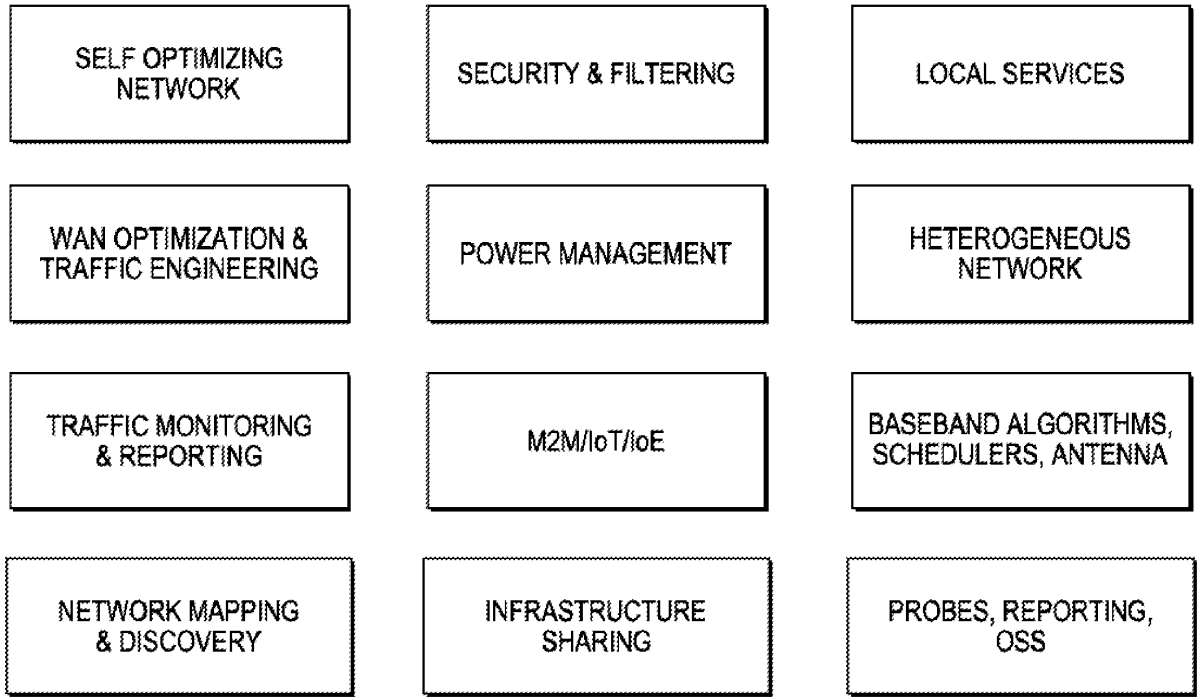

SELF OPTIMIZING NETWORK
SECURITY & FILTERING
LOCAL SERVICES
WAN OPTIMIZATION & TRAFFIC ENGINEERING
POWER MANAGEMENT
HETEROGENEOUS NETWORK
TRAFFIC MONITORING & REPORTING
M2M/IoT/IoE
BASEBAND ALGORITHMS, SCHEDULERS, ANTENNA
NETWORK MAPPING & DISCOVERY
INFRASTRUCTURE SHARING
PROBES, REPORTING, OSS

FIG. 7

FIG. 8

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0015

FIG. 9

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0016

FIG. 10

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0017

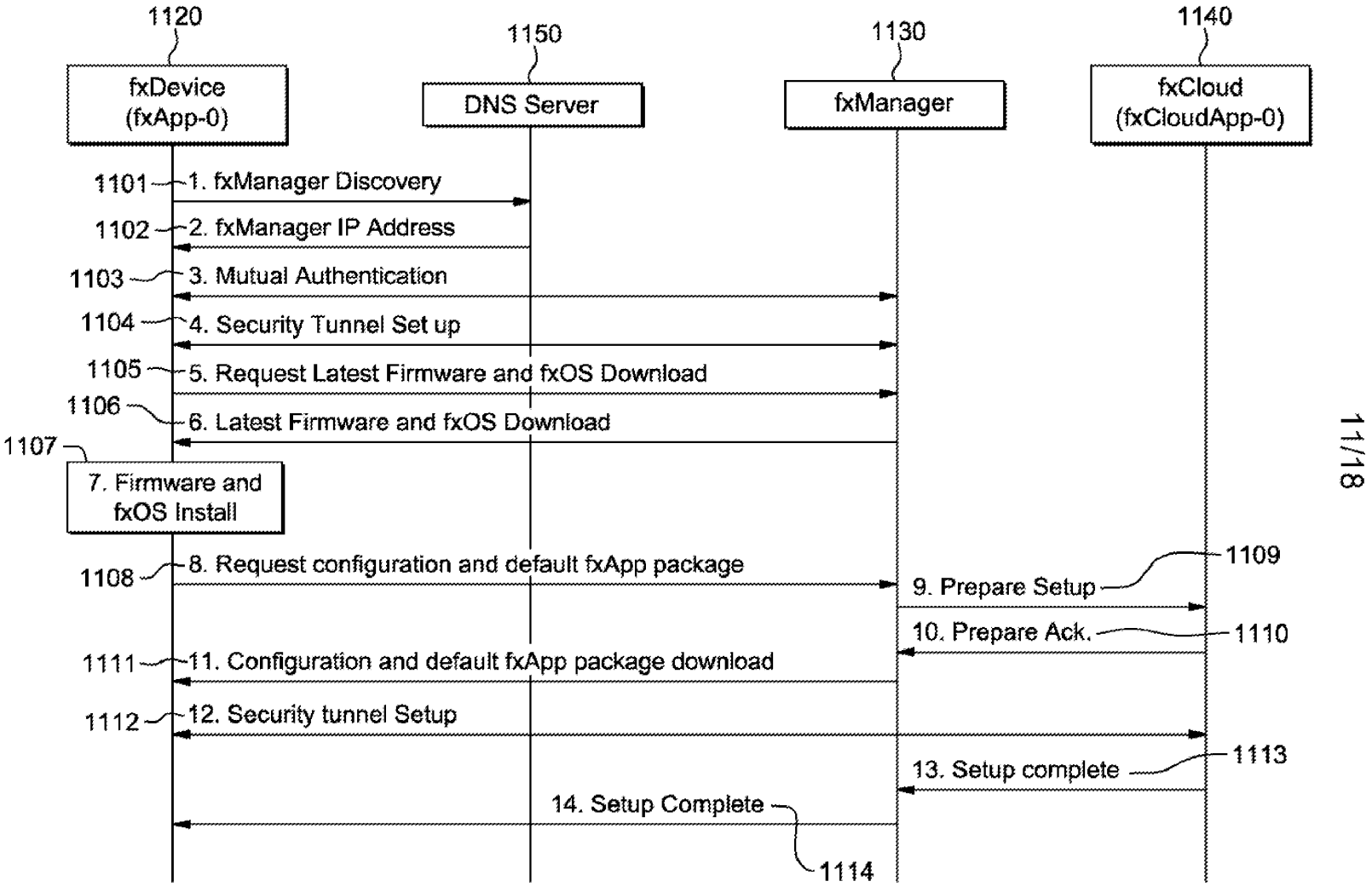


FIG. 11

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0018

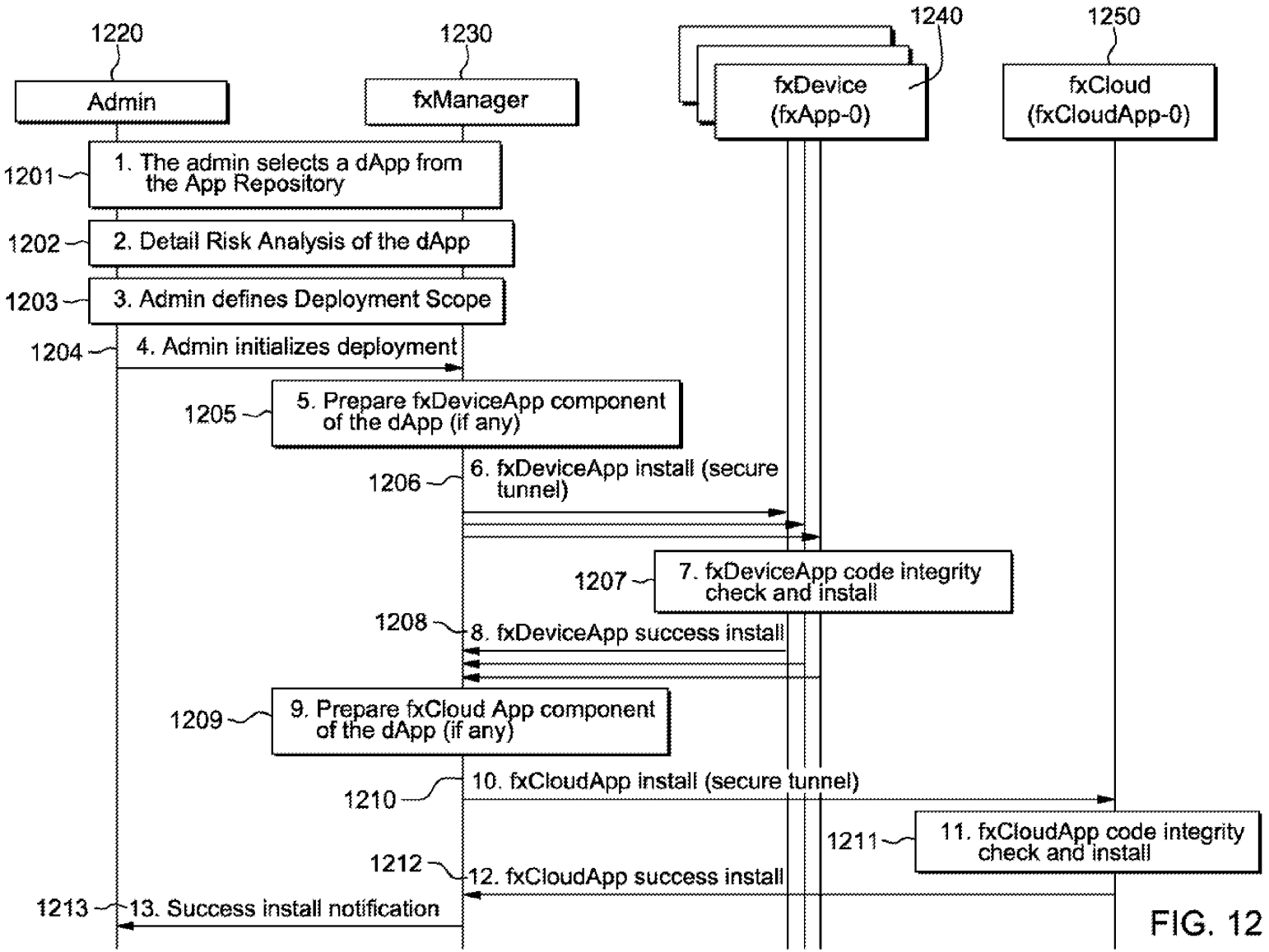

1220
1230
1240
1250
Admin
fxManager
fxDevice
(fxApp-0)
fxCloud
(fxCloudApp-0)
1201
1. The admin selects a dApp from the App Repository
1202
2. Detail Risk Analysis of the dApp
1203
3. Admin defines Deployment Scope
1204
4. Admin initializes deployment
1205
5. Prepare fxDeviceApp component of the dApp (if any)
1206
6. fxDeviceApp install (secure tunnel)
1207
7. fxDeviceApp code integrity check and install
1208
8. fxDeviceApp success install
1209
9. Prepare fxCloud App component of the dApp (if any)
1210
10. fxCloudApp install (secure tunnel)
1211
11. fxCloudApp code integrity check and install
1212
12. fxCloudApp success install
1213
13. Success install notification

FIG. 12

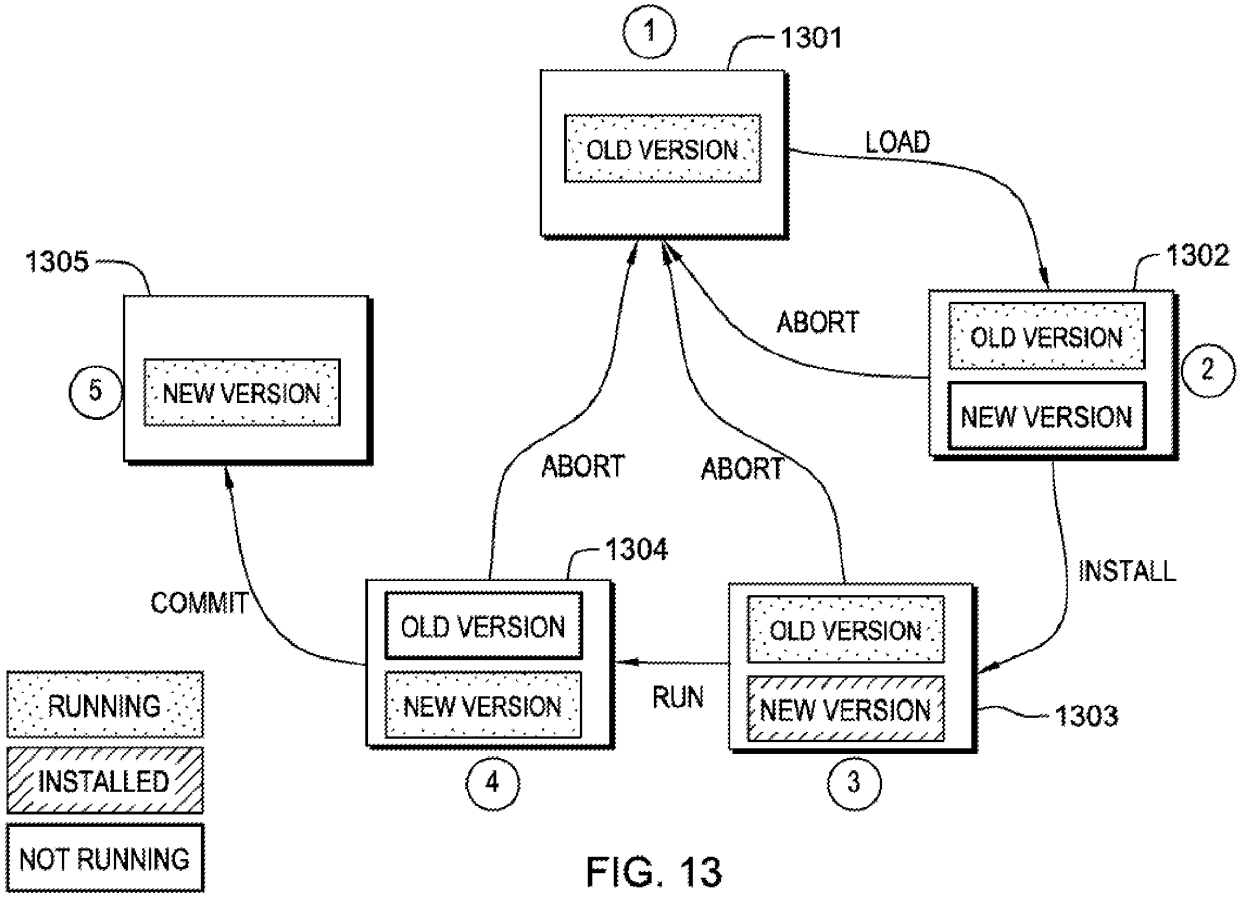

1
1301
OLD VERSION
LOAD
1305
1302
ABORT
OLD VERSION
5
NEW VERSION
2
NEW VERSION
ABORT
ABORT
1304
INSTALL
COMMIT
OLD VERSION
OLD VERSION
RUN
NEW VERSION
NEW VERSION
1303
RUNNING
INSTALLED
NOT RUNNING
4
3

FIG. 13

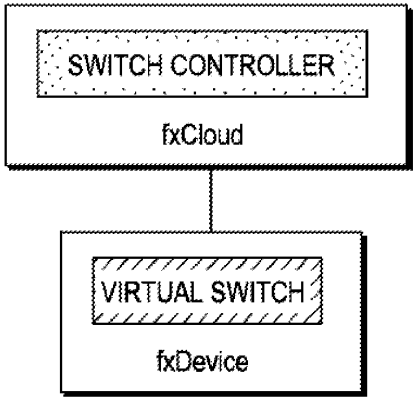


FIG. 14A

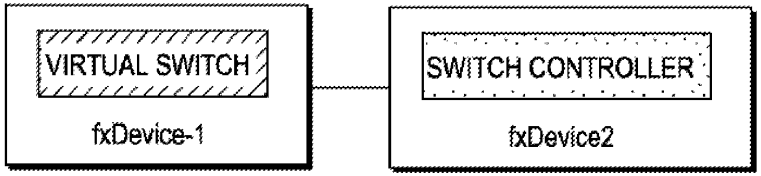


FIG. 14B

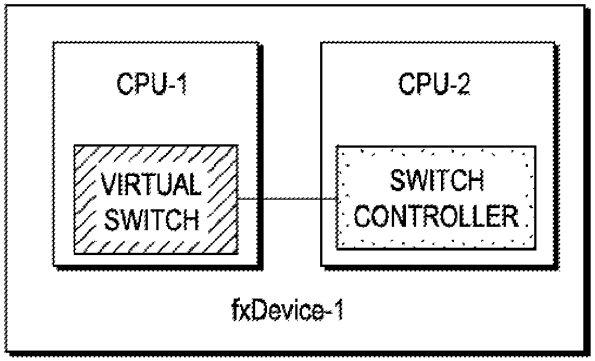


FIG. 14C

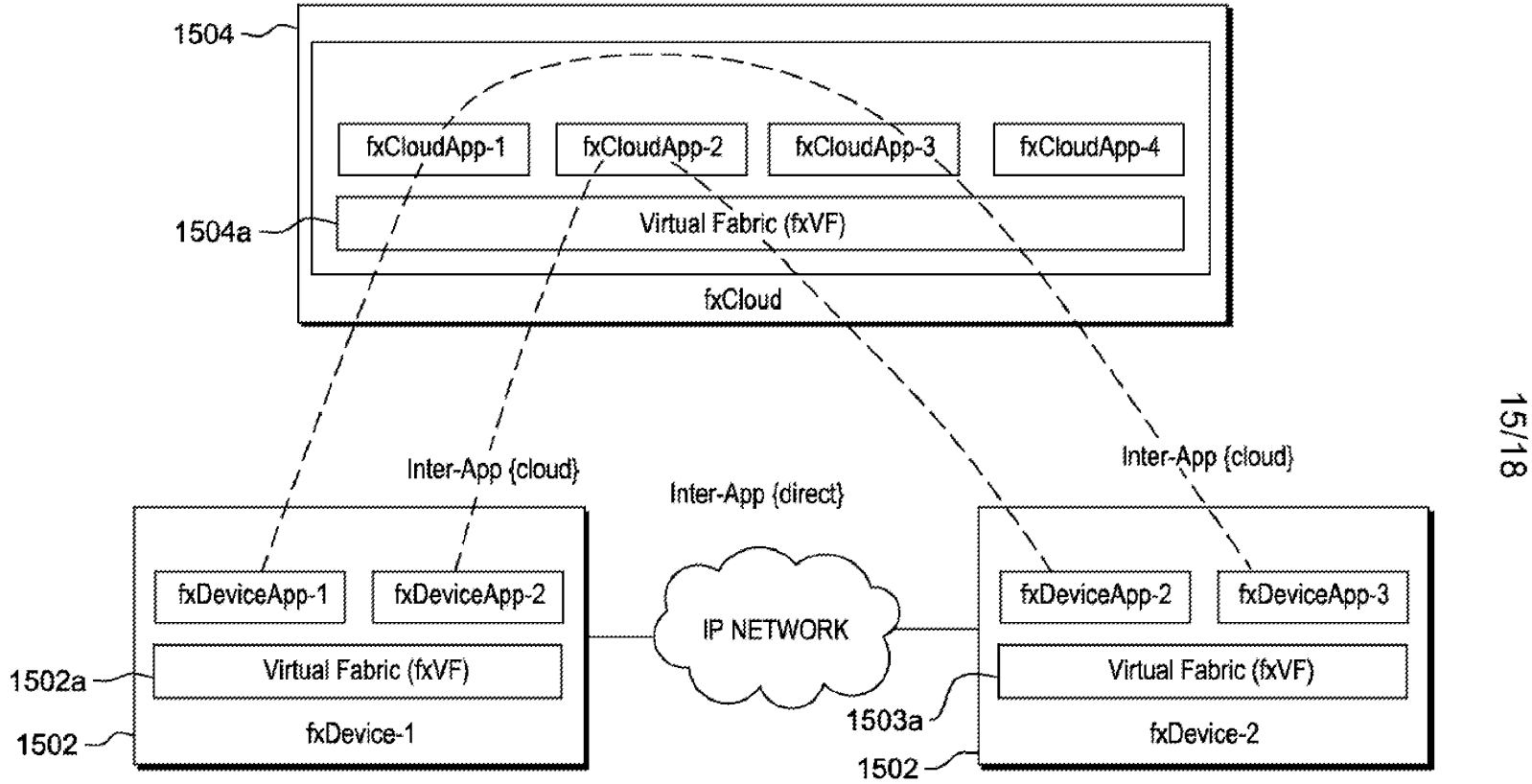

1504
fxCloudApp-1
fxCloudApp-2
fxCloudApp-3
fxCloudApp-4
1504a
Virtual Fabric (fxVF)
fxCloud
Inter-App {cloud}
Inter-App {direct}
Inter-App {cloud}
fxDeviceApp-1
fxDeviceApp-2
IP NETWORK
fxDeviceApp-2
fxDeviceApp-3
1502a
Virtual Fabric (fxVF)
Virtual Fabric (fxVF)
1503a
1502
fxDevice-1
fxDevice-2
1502

FIG. 15

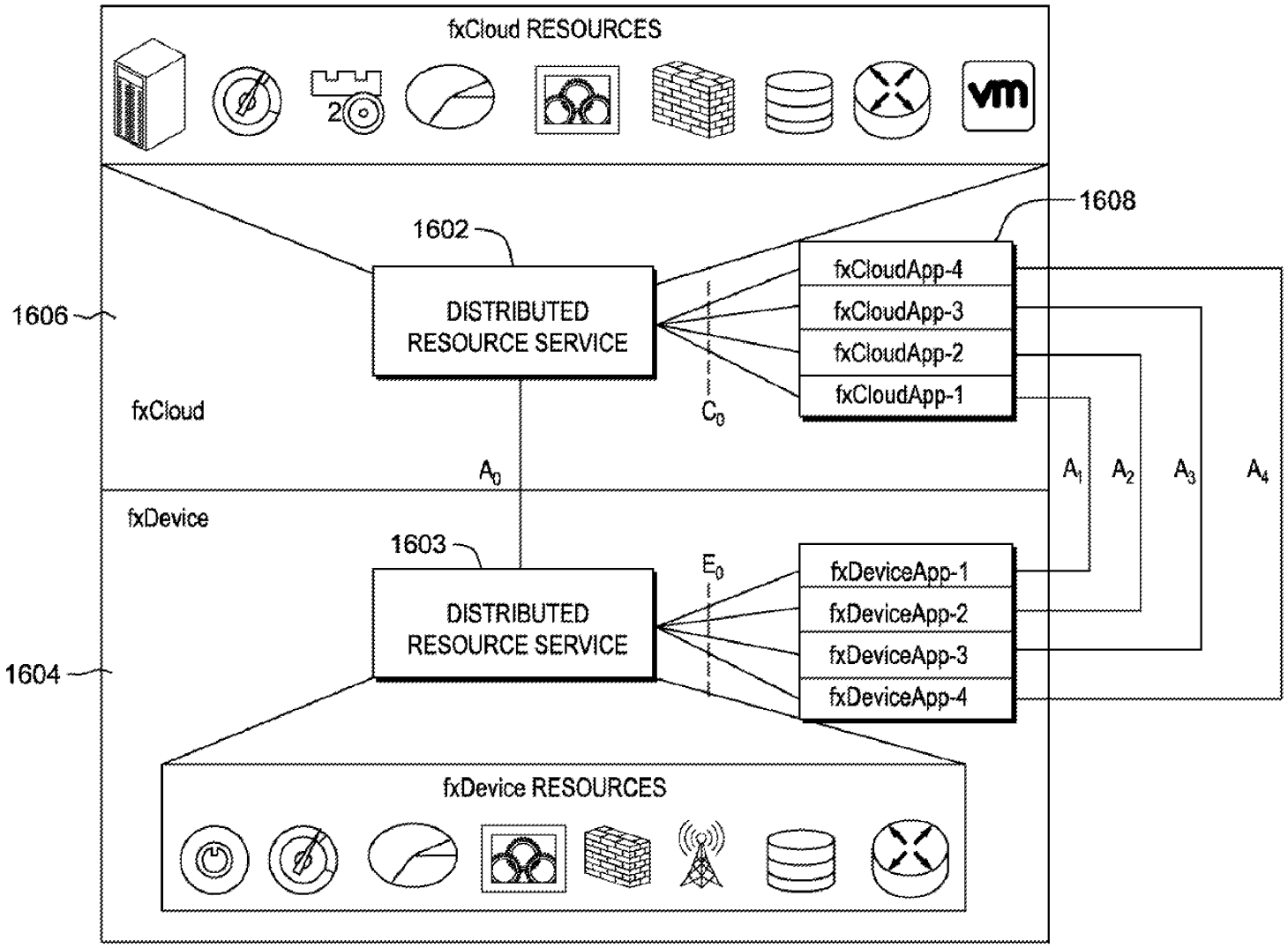

fxCloud RESOURCES
1602
1608
1606
DISTRIBUTED
RESOURCE SERVICE
fxCloudApp-4
fxCloudApp-3
fxCloudApp-2
fxCloudApp-1
fxCloud
C_0
A_0
A_1
A_2
A_3
A_4
fxDevice
1603
E_0
DISTRIBUTED
RESOURCE SERVICE
fxDeviceApp-1
fxDeviceApp-2
fxDeviceApp-3
fxDeviceApp-4
1604
fxDevice RESOURCES

FIG. 16

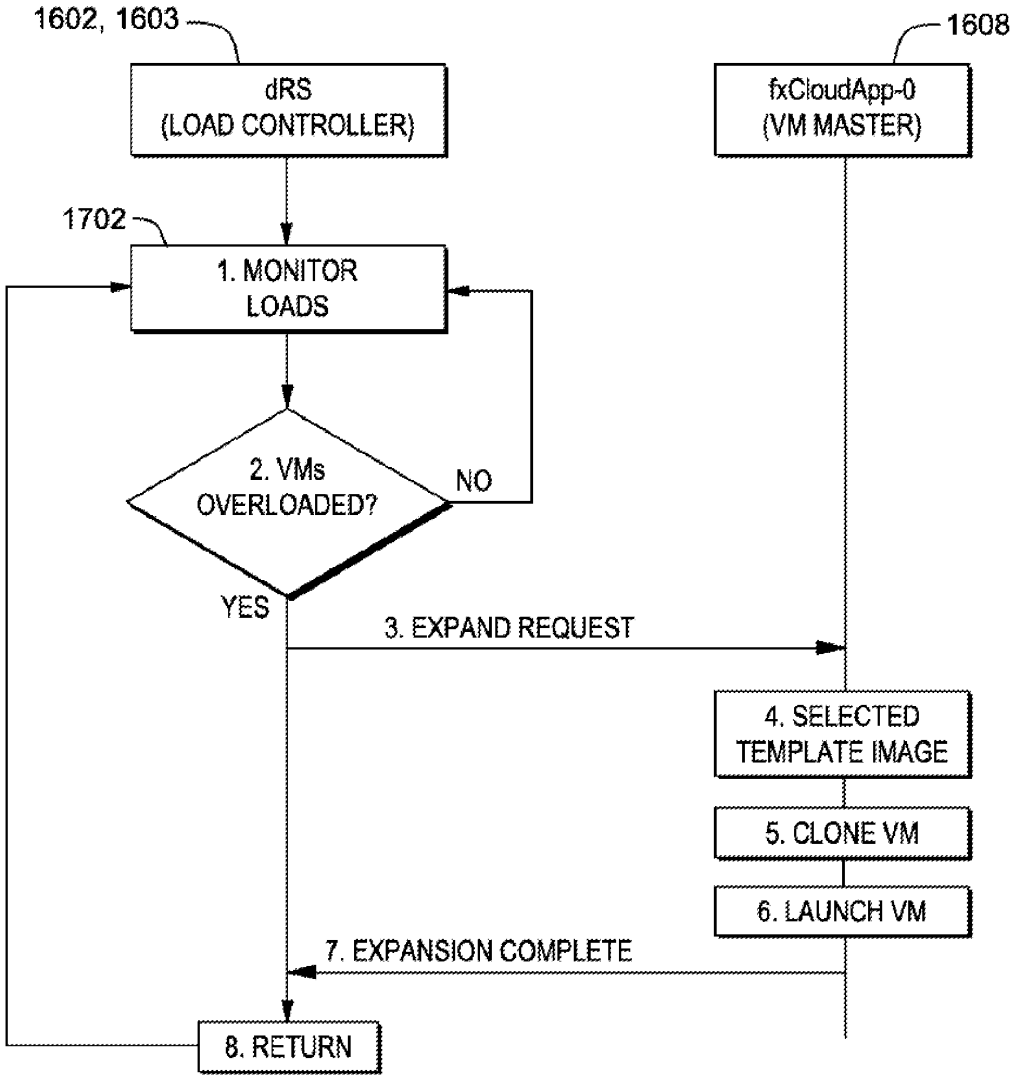

1602, 1603
dRS
(LOAD CONTROLLER)
1608
fxCloudApp-0
(VM MASTER)
1702
1. MONITOR LOADS
2. VMs OVERLOADED?
NO
YES
3. EXPAND REQUEST
4. SELECTED TEMPLATE IMAGE
5. CLONE VM
6. LAUNCH VM
7. EXPANSION COMPLETE
8. RETURN

FIG. 17A

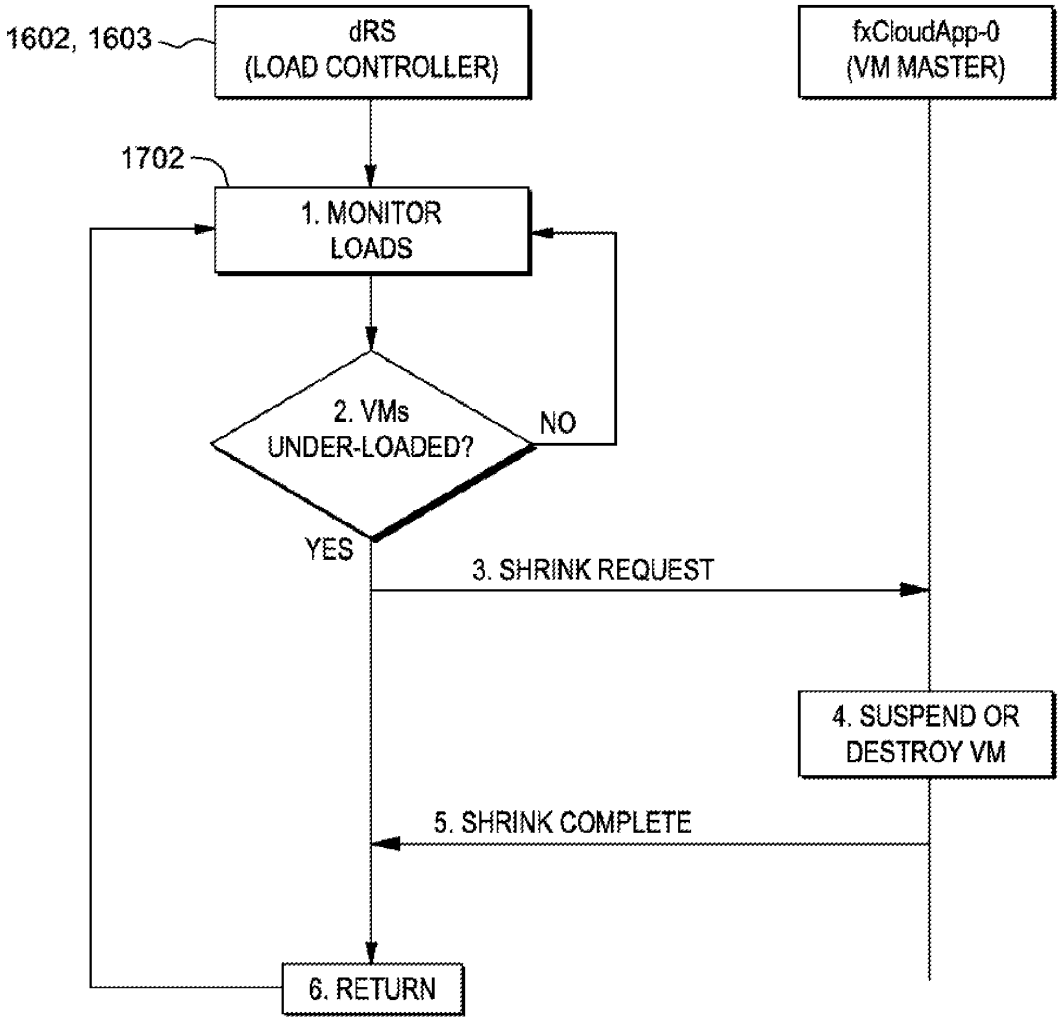

1602, 1603
dRS
(LOAD CONTROLLER)
fxCloudApp-0
(VM MASTER)
1702
1. MONITOR
LOADS
2. VMs
UNDER-LOADED?
NO
YES
3. SHRINK REQUEST
4. SUSPEND OR
DESTROY VM
5. SHRINK COMPLETE
6. RETURN

FIG. 17B

PTO/AIA/01 (06-12)
Approved for use through 01/31/2014. 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.

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

Title of Invention	Distributed Software Defined Networking

As the below named inventor, I hereby declare that:

This declaration is directed to:

- [■] The attached application, or
- [ ] United States application or PCT international application number _____

filed on _____.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

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.

LEGAL NAME OF INVENTOR

Inventor: Vivek Ramanna Date (Optional): June 2, 2014

Signature: Vivek

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form or must have been previously filed. Use an additional PTO/AIA/01 form for each additional inventor.

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.

PTO/AIA/01 (06-12)
Approved for use through 01/31/2014. 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.

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

Title of Invention	Distributed Software Defined Networking

As the below named inventor, I hereby declare that:

This declaration is directed to:

- [x] The attached application, or
- [ ] United States application or PCT international application number _____ filed on _____.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

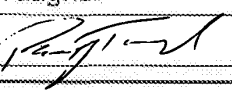
I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

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.

LEGAL NAME OF INVENTOR

Inventor: Pouya Taaghol Date (Optional): June 2, 2014

Signature: 

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form or must have been previously filed. Use an additional PTO/AIA/01 form for each additional inventor.

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.

PTO/SB/08a (07-09)
Approved for use through 11/30/2020. 0MB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENTOF COMMERCE
Under the Paperwork Reduction Act of 1995 ,no persons are reauired to respond to a collection of information unless it contains a valid 0MB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 4

Complete if Known	
Application Number	
Filing Date	
First Named Inventor	Taaghol
Art Unit	
Examiner Name	
Attorney Docket Number	TAAG001CONT3

U.S.PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 7114021	9-2006	Seshadri	
		US- 7630706	12-2009	Weigand	
		US- 8132166	3-2012	DeHaan	
		US- 8528108	9-2013	Ammer	
		US- 8650299	2-2014	Huang	
		US- 8756595	6-2014	Arcilla	
		US- 8843933	9-2014	Holler	
		US- 9226333	12-2015	Steiner	
		US- 20020016954	2-2002	Charisius	
		US- 20040107417	6-2004	Chia	
		US- 20040237081	11-2004	Homiller	
		US- 20050246703	11-2005	Ahonen	
		US- 20070078988	4-2007	Miloushev	
		US- 20080209491	8-2008	Hasek	
		US- 20090007091	1-2009	Appiah	
		US- 20090113327	4-2009	Pogrebinsky	
		US- 20090276771	11-2009	Nickolov	
		US- 20090300057	12-2009	Friedman	
		US- 20110075611	3-2011	Choi	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. Applicant is to place a check mark here if English language Translation is attached.

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 2 hours 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, 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, ca/11-800-PTO-9199 (1-800-786-9199) and select option 2.

PTO/SB/08a (07-09)
Approved for use through 11/30/2020. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no persons are reauired to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 2 of 4

Complete if Known	
Application Number	
Filing Date	
First Named Inventor	Taaghol
Art Unit	
Examiner Name	
Attorney Docket Number	TAAG001CONT3

U.S.PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 20120300615	11-2012	Kempf	
		US- 20130007254	01-2013	Fries	
		US- 20130160147	06-2013	Draluk	
		US- 20130263104	10-2013	Basel	
		US- 20140052976	02-2014	Marino	
		US- 20140241158	08-2014	Anthony, Jr.	
		US- 20140254373	09-2014	Varma	
		US- 20140259012	09-2014	Nandlall	
		US- 20140317293	10-2014	Shatzkamer	
		US- 20150173121	06-2015	Miklos	
		US- 20110265077	10-2011	Collison	
		US- 20120066665	03-2012	Kunze	
		US- 20120096071	04-2012	Murphey	
		US- 20130223442	08-2013	Narayanan	
		US- 20150160877	06-2015	Larkin	
		US- 20150242237	08-2015	Spivak	
		US- 20160224460	08-2016	Bryant	
		US- 20190165997	05-2019	Shaikh	
		US- 8843933	09-2014	Holler	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. Applicant is to place a check mark here if English language Translation is attached.

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 2 hours 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, 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, ca/11-800-PTO-9199 (1-800-786-9199) and select option 2.

PTO/SB/08a (07-09)
Approved for use through 11/30/2020. 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.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 3 of 2

Complete if Known	
Application Number	
Filing Date	
First Named Inventor	Taaghol
Art Unit	
Examiner Name	
Attorney Docket Number	TAAG001CONT3

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 8495611	07-2013	McCarthy	
		US- 20120266156	10-2012	Spivak	
		US- 20060271341	11-2016	Brown	
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T[6]

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

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 2 hours 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, 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 (1-800-786-9199) and select option 2.

PTO/SB/08b (07-09)
Approved for use through 11/30/2020. 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.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 4 of 4

Complete if Known	
Application Number	
Filing Date	
First Named Inventor	Taaghol
Art Unit	
Examiner Name	
Attorney Docket Number	TAAG001CONT3

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	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, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		McKeown, et al. "OpenFlow: Enabling Innovation in Campus Networks", March 14, 2008.	
		Li et al. "Toward Software-Defined Cellular Networks", "2012 European Workshop on Software Defined Networking".	
		"Network Functions Virtualisation", "SDN and OpenFlow World Congress", Darmstadt-Germany, October 22-24, 2012.	
		Yap et al., "OpenRoads: Empowering Research in Mobile Networks", ACM SIGCOMM Computer Communication Review, 2010.	
		Bansal et al., "OpenRadio: A Programmable Wireless Dataplane", HotSDN'12, August 13, 2012, Helsinki, Finland.	
		Stallings et al., "Software-Defined Networks and OpenFlow", The Internet Protocol Journal, vol. 16, No. 1, March 2013.	
		Luo et al., "Software Defined Networking across Distributed Datacenters over Cloud", IEEE Int. Conference on Cloud Computing Tech, 2013	
		Chappell, David, "A Short Introduction to Cloud Platforms", Aug. 2008.	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.
This collection of information is required by 37 CFR 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 2 hours 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, 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 (1-800-786-9199) and select option 2.

2013 IEEE International Conference on Cloud Computing Technology and Science

Software Defined Networking across Distributed Datacenters over Cloud

Mon-Yen Luo and Jun-Yi Chen
Department of Computer Science & Information Engineering,
National Kaohsiung University of Applied Sciences, Taiwan
* myluo@kuas.edu.tw

Abstract- **Cloud computing, in particular its system spread across several data centers, poses a unique set of networking challenges and management problems. This paper presents our design, implementation, and experience in tackling these challenges by using the concepts and technologies of software-defined network (SDN). We define a network abstraction to incorporate the physical and virtual data plane within a datacenter. We propose a new network primitive to address the interconnection issues among multiple datacenters. A policy plane consolidating these mechanisms provides a programmable interface to enable a wide range of policies and new networking mechanisms. We present the performance result on a real deployment over the production networks. The experience and initial performance results demonstrate that the SDN is a promising solution to cloud networking.**

Keywords: Software-Defined Network; Datacenter Network; Cloud Federation; OpenFlow.

I. INTRODUCTION

Cloud computing has emerged as a promising paradigm for cost efficient and pervasive service delivery across data communication networks. While the computing and distributed system issues of cloud technologies have been extensively investigated, fewer attention has been devoted to the unique set of networking challenges to a cloud system. In order to support multi-tenancy and provide more flexible management of resources, the datacenters usually use virtualization technologies (i.e., hypervisors) to abstract the physical resources. The dynamic location of service deployments and virtualization of hardware resources are stressing the network and protocols. Cloud networks must support automated management (creation, deletion, and migration) of enormous numbers of virtual machines (VMs), large numbers of attached physical and virtual devices, and isolated independent subnetworks for multi-tenancy (application components belonging to separate different tenants are collocated on the same physical substrate) [1]. Recent research in cloud networks has mainly focused on networking mechanisms inside the data center [2]. Little attention has been paid to the networking mechanisms for cloud whereby data centers are interconnected through the Internet [2, 3].

Under the support of a project awarded by the National Science Council of Taiwan government, we are currently designing and maintaining a network testbed that interconnects several heterogeneous datacenters of our research partners. Our current implementation and deployment has supported several research projects and applications, including cloud federation [4], international testbed interconnection [5], and network/security research experiments over production networks [6]. We faced some challenges that also exist in some international experimental infrastructures (e.g., GENI [7, 8], and FIRE [9]), cloud federation, or any cloud system spread across multiple datacenters. This paper describes our design, implementation, and experience in tackling these challenges by using the concepts and technologies of software-defined network (SDN). SDN is emerging as a promising solution to make computer networks more programmable, enable innovation in network design, and lower the barrier to deploying new services [10, 11]. The core concepts of SDNs are (i) separating the control (policy) plane from the data (forwarding) plane, (ii) providing well-defined forwarding abstractions (e.g., flow tables in switches) with standard programmatic interfaces (e.g., OpenFlow [12]), (iii) and exposing high-level abstractions to implement a wide range of network policies and new network technologies. Recently, researchers have made quantified arguments that SDN is a viable approach to single datacenter networking [13], enterprise networks [14], or even wide-area traffic management [15].

The major contribution of this paper is toward the design of a system to transform networks among a collection of independent autonomous datacenters into a manageable, flexible, multitenant transport fabric. We begin with an overview of background material on datacenter networking and SDNs, also describing our deployment environment and its networking challenges (Section II). Generally, the challenges arises from the multitenant nature of cloud computing. Each tenant service should be allocated a portion of network resources (we refer to this as a virtual network in this paper) decoupled from the underlying physical equipment. In our deployment, VMs belonging to a virtual network may be hosted either on the hosts within a datacenter or on a collection of physical hosts across different datacenters. We require a system that creates and manages virtual networks on-demand across multiple datacenters. We discuss the design issues and present a high-level overview of our design in Section III, followed by a detailed description in Section IV. We provide the results of performance measurement from the implementation, based on real production networks (Section V). The initial performance data show that our system can create virtual networks across datacenters efficiently. We discuss and conclude our work in Section VI.

II. BACKGROUND AND RELATED WORK

This section first describes background information and related work on data center networks. We then discuss the

978-0-7695-5095-4/13 $31.00 © 2013 IEEE
DOI 10.1109/CloudCom.2013.87

IEEE computer society

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0032

requirements resulting from a system across distributed datacenters and current related solutions.

A. *Datacenter Network*

Data centers exist in various types. Some data centers, for example, use commodity computers running Linux without a hypervisor to process data on a large scale (e.g., Google's datacenters). In this paper, we focus on another popular type of datacenters (i.e., the data centers comprise many racks of commodity servers using hypervisors to support multiple VMs per host). Machines in such datacenters are typically organized into racks and rows under a logical hierarchical network tree [2]. Each rack contains several machines interconnected by a top-of-rack (ToR) switch, which serves as the leaves of the tree and delivers high bandwidth to directly connected hosts. Switches at the leaves have a limited number of high-speed (e.g., 10 GigE) uplinks to one or more network devices that aggregate and transfer packets among the leaf switches. At the root levels of the tree, there are core switches with significant high throughput and switching capacity to relay traffic for inter-row communication. Many proposals and commercial products target on searching for more efficient and low-cost data center network fabrics, such as FatTree [16, 2], and Clos topologies [17, 18, 2]. These topologies face a similar set of challenges, such as improving cost-effectiveness (reducing oversubscription rate), supporting high bisection bandwidth, providing guaranteed bandwidth for each tenant, and increasing fault tolerance.

Much of the research for cloud networking has focused on developing network infrastructure and embedding algorithms optimized for particular topologies. PortLand [19] used a hierarchical pseudo MAC address (PMAC) for all end hosts (or VM) to encode their positions based on a fat tree topology. The edge (ToR) switches, to which a server hosting a VM is connected, perform the rewriting between actual MACs (AMAC) and PMACs. The PMAC design enables efficient and loop-free forwarding (PortLand uses the longest prefix-match), and reduces forwarding states in switches. A fabric manager (a process running on a dedicated machine) maintains a full connectivity matrix of switches and responds to ARP resolution, preventing broadcast flooding. VL2 [20] uses a Clos network topology and implements Valiant Load Balancing (VLB) by sending traffic to random intermediate switches. In VL2, all physical machines and services have application-specific IP addresses (AA), and all switches are assigned a location-specific IP address (LA) with a forwarding mechanism based on OSPF. Similarly, a centralized address manager (AM) maintains the AA-to-LA address mappings. An IP-in-IP encapsulation mechanism is implemented to route on LAs but deliver packets to AAs. NetLord [21] leverages a multipath forwarding scheme termed SPAIN [22] for datacenters using commodity switches. NetLord designs an agent in the hypervisors to encapsulate, route, de-capsulate, and deliver all packets from and to the local VMs. NetLord agents collaborate with central configuration repositories to gather and maintain all the information needed for the encapsulation.

The aforementioned discussion is definitely not a comprehensive list, but represents some important and common characteristics for data center networks. These characteristics led to our design adopting SDN architecture, which will be discussed in Section III.

B. *Issues Related to Distributed Datacenter*

Further challenges exist in a cloud system spread across multiple datacenters, while a virtual network of one particular tenant service may be deployed across (or migrated temporarily to) several datacenters. First, it is very possible that each datacenter has its own addressing scheme and forwarding fabric, which complicates the deployment of a virtual network spread across multiple datacenters. Second, the network infrastructure between each datacenter usually is shared with public Internet services. It is challenging to decouple resources from the public shared Internet facilities to support various tenant services. The traffic among the network interconnection of our testbed can be categorized into three classes: (i) regular Internet traffic, (ii) interconnection traffic between federated clouds (e.g., VM migration, or interaction of service components deployed across multiple datacenters), and (iii) experimental traffic from various research projects. As a result, the third challenge is that we must isolate production and the private cloud network from each other.

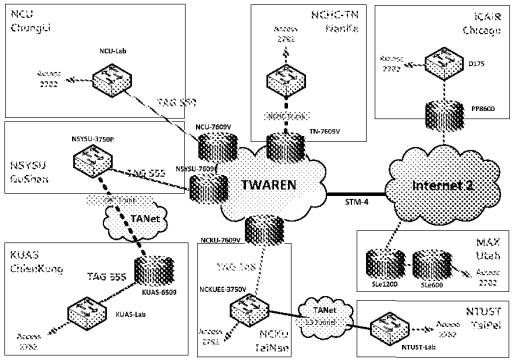


Figure 1: Network Topology of Current Deployment

Let us use current deployment of our testbed to illustrate these issues. As depicted in Figure 1, datacenters in our testbed are majorly interconnected with shared, public layer-3 networks (i.e., the Taiwan Academic Network (TANet) and Taiwan Advanced Research and Education Network (TWAREN)). With supports from iCAIR at Northwestern University and the National Center for High-Performance Computing (NCHC), our testbeds have also been extended to connect with iGENI project [5] in the United States. To isolate the traffic, some of our partner universities (e.g., NCKU, KUAS, NCU) applied some VPLS [23] paths to interlink the private cloud systems. However, VPLS only offers a partial solution to the aforementioned problems. The first reason for this shortcoming is that the use of 12-bit VLAN IDs limits the solution's scalability. At first, in our

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0033

environment, we are only allowed to use the VLAN 2782 to interconnect all sites, which causes another serious scalability and performance problem. The system suffers from issues caused by flooding mechanisms (e.g., ARP broadcast), the huge MAC table size, and the use of Spanning Tree Protocol in a huge layer-2 network [24]. Third, the VLAN, or VPLS, does not have an easy management way to configure the path across several network domains.

We require a system to create and manage simultaneous virtual networks on-demand with arbitrary topologies on loosely coupled distributed datacenters. Some researchers have tried to use IP tunneling techniques and overlay network concepts to extend virtual networks across operational networks, such as VINI [25], VIOLIN [26], and Trellis [27]. In contrast to these systems, our work makes network virtualization coherent in the hypervisor and with the support of NetFPGA, resulting in a good performance and isolation. CloudNet [28] relies on provider-based virtual private networks (VPNs) to extend private network across data centers. The CloudNet is mainly designed for VM live migration, thus, it also implements optimizations for VMs to migrate across the wide area network (WAN). However, the VPN-based mechanisms are mainly point-to-point dedicated connectivity and require a series of interactions and manual operations between network operators and service providers. By contrast, we propose a novel mechanism to extend virtual networks by dynamically organizing and utilizing the VPLS paths (i.e., VLAN tag). The performance result will show that this approach can set up paths to extend a virtual network in only a few seconds while incurring very little overhead in operation.

III. . SYSTEM DESIGN

In this section, we first review the key concepts of SDNs, and then discuss how these principles lead to our design. We then give an overview of our system architecture.

A. *Sofware Defined Networks*

SDN is part of a long history of efforts to make data networks more programmable, which aims to make the network more manageable, enables innovation in network design, and lowers the barrier to deploying new services. In general, SDN has three principal concepts. First, SDN decouples the control plane (which decides how to handle the traffic) and the data plane (which forwards traffic according to decisions that the control plane makes) from each other. Second, SDN logically consolidates the control planes to offer a vision of unified control and management over the states in the network's data-plane elements. Third, SDN provides a well-defined application programming interface (API) to enable the creation of control policies or new applications. OpenFlow [13] is a widely popular instantiation of SDN principles. The OpenFlow defines a data-plane model and a control-plane API by building on technology that commodity switches already supported. The creation of the OpenFlow API was followed quickly by various implementations of controller platforms (e.g., POX [29], Beacon [30], Floodlight [31]) that control the OpenFlow switches and enable new applications. We use OpenFlow in our current implementation and as a concrete instance of SDN throughout the paper. However, we believe that different SDN APIs could be used with our proposed system.

The data-path of an OpenFlow switch is controlled by a table of packet-handling rules. Each rule may contain a pattern (defined by some part of the packet header) that matches a subset of traffic, a list of actions (e.g., drop, flood, forward out a particular interface, modify a header field, send the packet to the controller) on the traffic that matches this rule, a set of counters (to track the number of bytes and packets), and a priority. Upon receiving a packet, an OpenFlow switch identifies the highest-priority matching rule, performs the associated actions, and manipulates the counters. An OpenFlow switch can be programmed (i.e., install forwarding rules) by only one controller (called *master* controller) in either proactive or reactive mode. The controller may proactively install some rules to some of the managed switches. In the reactive mode, an OpenFlow switch sends the first packet from traffic that does not match any rule to a controller, triggering the controller to insert the appropriate flow entries into the table of the OpenFlow switch.

While an important design goal of our system is practically deployable, the SDN architecture provides a practical approach to building a system from the existing commodity switches and real production networks. Based on the observations in Section II, we think that we can exploit SDN architecture to meet our system requirements in the following aspects:

- Many researchers have proposed some important and practical schemes within a single datacenter network. The SDN architecture enables us to extend the network abstraction and design a system architecture that can be compatible with these existing approaches.
- In a multitenant environment like cloud, customers may want the network for their service to be isolated (e.g., a private L2 broadcast domain) from others. This requirement is particularly challenging if such a virtual network is deployed across datacenters. We design a novel mechanism to address this challenge, which can be programmed by an SDN-based architecture.
- The cloud network is still in its infancy, it is therefore important for a management system to provide network control that is flexible enough to support a variety of requirements, and it is extensible to meet a wide range of future needs. The SDN architecture is a promising approach to provide a logically centralized programming environment based on a set of network abstractions.

B. *System Overview*

Our system serves multiple datacenters, each with a number of server racks and network devices. The system architecture can be logically viewed in three layers of software stack, depicted in Figure 2, providing the hooks to our previous work

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0034

or other middleware and serving as a control plane to orchestrate all operations of virtual networks. First, we abstract the data plane of both the physical and virtual switches and synergize them coherently to enable the dynamic configuration and management of virtual networks within a datacenter. We design a software module that resides in the kernel or hypervisor of each host in a datacenter. We also extend a POX [29] controller with some added function to orchestrate the software modules to create a distributed data plane layer.

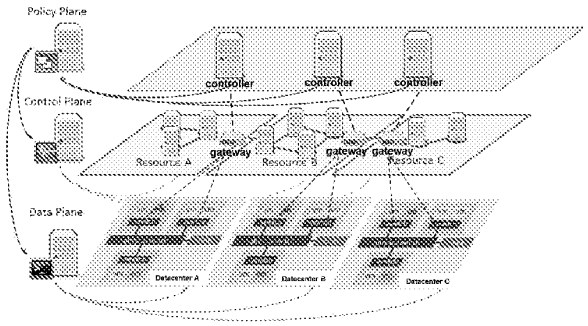


Figure 2: System Overview

The second layer is designed to deal with the interconnection issues among multiple datacenters. The major building blocks of this layer include a gateway switch that is designed based on the OpenFlow protocol. Each datacenter should at least have one gateway system located on the edge of its network to relay traffic across sites. The operation of the gateway system is based on a novel mechanism termed virtualtransit [32]. We assume that each pair of datacenters may set up one or more forwarding paths such as VPLS or MPLS. The basic idea of the virtualtransit is to abstract these paths and dynamically select a suitable path to relay the traffic to the selected datacenter. A distributed control plane layer provides inter-datacenter mechanisms to create and manage virtual private networks spread across public Internet.

The third layer is a logically central control framework containing the global network topology and the policy that determines the behavior of the whole system. The control framework performs route computation on the virtualtransit and explicitly installs flow information to the switches along the chosen route. The control framework consists of multiple controllers that can be replicated for redundancy, and its table can be partitioned for performance scalability. Consolidating all flow decisions to the controller framework, a central service management system can be implemented to enforce a policy throughout the whole system. The system can also be extended further to add new functions and features by simply updating the controller. For instances, our control framework provides interface to allow integration with other control frameworks such as ORCA [33] or our previous control framework [4] for user authentication, or resource registration and discovery.

IV. Implementation and Deployment

Putting all the three layers together, we describe how our system works in more detailed in the following sections. We also describe in some detail our implementation and deployment.

A. *Data Plane Layer*

From the perspective of SDNs, the basic idea of our work in the data plane is to design an appropriate network abstraction to incorporate various addressing and forwarding schemes, so that we can enable a virtual network to be deployed across datacenters in a coherent way. We abstracted the virtual network as a switching element at the core, with a set of network primitives such as port, linking, and policy. In our previous work [3], we implemented a software module that can reside in the hypervisors of host systems to enable the creation and management of virtual networks. In this paper, we further extend our previous work to enable a software-defined networking architecture. We extended the generic control interface of the software module as follows:

- Switching element: This module implemented APIs for creating, removing, and monitoring a virtual switch in a server node. In an SDN architecture, we need a bootstrapping function to make the virtual switch know the controller's location and register itself to the controller. Our system reserves a control channel (VLAN 0 in current implementation) to let switches find and bootstrap connectivity to the controller. Once a connectivity is established, the controller recognizes this switch and aggregates this information to construct the network topology in an information database.
- Virtual port: This module provides functions for adding a virtual port to a VM, removing a virtual port, or disassociating a port from a VM. We also implemented some address assignment mechanisms in this module, which can be customized to support different addressing schemes. We further implement some functions to query and send the link-state information to controller to complete the network topology database.
- Linking: This module provides functions for creating, modifying, and removing connection paths between virtual ports. In our SDN architecture, the controller can configure the virtual switches to perform packet forwarding in three different ways. The first one is called autonomous mode. In current implementation for autonomous mode, we implement two prominent packet-forwarding schemes: SPAIN [22] and VL2 [20]. The autonomous mode is suitable for tenant services that do not require specific topology. The second mode is called proactive mode, which is suitable for tenant service that requires specific topology. For example, recent research has pointed out that the middlebox (e.g., firewalls, network address translators, server load balancers) may further complicate the challenges of managing the datacenter network infrastructure. For some enterprise

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0035

applications, it is necessary to specify the data flow and topology for their networks. We implement functions that are used to create a topology for a given virtual network, including layer-2 mechanisms (e.g., VLAN, VPLS) and layer-3 mechanisms (e.g., IP tunneling, GRE [34]). The third mode is reactive mode, the behavior of which is defined by the controller.

- Policy: This module provides functions for setting a path constraint, such as QoS constraints. These function libraries were designed to allow for further extension to support network management policies such as bandwidth allocation/management mechanisms.

Our mechanisms for intra datacenters do not dictate any particular networking topology or specialized switches. As a proof of concept of system flexibility, we have enabled three datacenter networking architectures with our system. If the intermediate switches are all OpenFlow-enabled switches, our system can support implementations of VL2 or PortLand as in the work [13]. For a datacenter that only uses commodity switches for intermediate switches, our system can use SPAIN by orchestrating the virtual switches and physical switches.

B. Control-Plane: Inter-datacenter Mechanisms

In our latest work [32], we have proposed a novel mechanism to address the interconnection issues among multiple datacenters. Our system does not require any modification to the switches in the datacenters. We add only some OpenFlow switches (OFS for short) that serve as a gateway system for the interconnection of datacenters. The data plane of OFS remains, and only processes packets according to the flow table, which is configured by the controller. The concept of virtualtransit is novel and promising because we can abstract the interconnection problem by adding a new network primitive that a single system can control dynamically. In this work, we further extend and generalize this mechanism by introducing the SDN architecture. Based on the network topology information, the controller framework manipulates a flow table with two parts (i.e., ARP table and transit table) to execute the virtualtransit. We use Figure 3 to show how the mechanism could operate in a real deployment. When a virtual network must be deployed across datacenters, an offline algorithm is invoked to compute the transit paths and configure the related information into the transit table. The major challenge of the inter-cloud mechanism is how to relay necessary information and isolate the local traffic of each datacenter. The controller framework manipulates the ARP table to address this issue. The reader is referred to [32] for further details on the operation of virtualtransit.

As a proof of concept, we use one single controller in the work described in [32]. While our measurements in Section V show that a single controller system may be able to manage thousands of machines, it also suggests that the network latency is a considerable concern while the gateway system needs to connect the controller across wide area network. As a result, multiple controllers forming a logically single control platform is desirable to provide high-availability or to scale to more datacenters over wide area networks. Our design of a distributed controller takes advantage of two properties of system behaviors and workload characteristics. First, the flow tables can easily be partitioned to several locations because the physical network topology (e.g., the possible transit path) is the same for all replicas, and its placement and binding can be handled in a purely distributed fashion without a strong consistency requirement on table content. Second, similar to the observation in an enterprise network [14], changes to the network state happen on very slow time scales relative to network flow arrivals. Thus, a standard replication strategy is enough to ensure the correctness of forwarding on each site and maintain consistency in global. As a result, we now use a full replication strategy for the flow table on each controller. Only some events (such as topology relocation, link failure, and change of transit path) result in a change of the controller state and incur overhead for consistency management. We expect future exploration on state partition or fault tolerance to make the system more efficient and robust

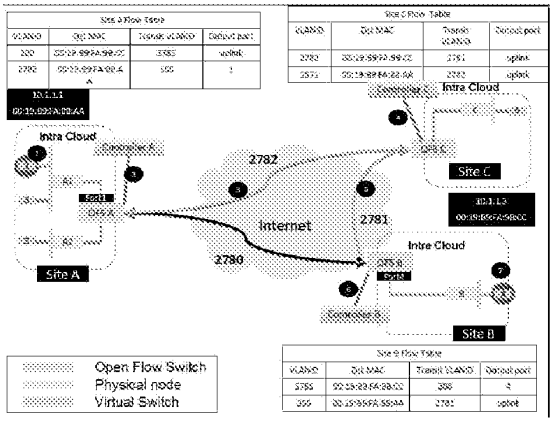


Figure 3: Operation of Inter cloud Mechanisms

C. Policy Plane

Based on the aforementioned two layers, we now can provide a logically centralized control plane that offers the programmability required to dynamically configure and manage the whole network. Due to the space limitation, we only discuss how a policy can be enforced throughout the system as a use case. We leave the comprehensive discussion to our future publications. Let us use Figure 3 to illustrate this use case. Figure 3 represents a real deployment example in our federated datacenter platform. In this example, five VMs (VM1 to VM5) belonging to a single tenant were allocated to different sites. System administrator can use the policy plane to specify the topology of this virtual network and the management policies for this network. The virtual network is then deployed to the network environment as depicted in Figure 3. The subnetwork in

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0036

site A used VLAN 100 for intra cloud communication, and the sub-network in site C used VLAN 200. Such a design is scalable because each datacenter may have complete and independent VLAN space, and an overlapping VLAN tag space is allowable. In contrast, the VPLS or VLAN in WAN environment is more expensive to deploy and manage. The virtualtransit is designed to solve this problem. A path computation mechanism is invoked to select a suitable path. In the example illustrated in Figure 3, that virtual network eventually was deployed using VPLS path 2782 (from Site A to Site C) and VPLS 2781 (from Site C to Site B).

Based on the topology requirement and path selection, the policy plane then parses the topology-related information and configures the ARP table and transit table. When the ARP query for VM4 is forwarded to OFSA, OFSA will consult the ARP table for the mapping, and then serve as the ARP proxy to respond to the ARP query. Such a design prevents ARP flooding and makes the system scalable and efficient. The module also updates the tables when one VM is migrated or destroyed because the VM creation, migration, and destruction are controlled by the management system or cloud manager. This confirms that an SDN architecture can support VM live migration efficiently by controllers, which install flow entries in all switches along the new path and inactivate (or remove) flow entries from the switches in the previous path upon receiving notification that the VM is moving.

When the first packet (after ARP query and reply) of the traffic flow from VM1 to VM4 arrives at the OFSA, the OFSA sends a control packet to the controller because it is missing in the local flow table. The information (OSF ID, VLAN ID, and Destination MAC) from the control packet is used as keys to search the mapping entry in the transit table. If a mapping entry is found, the transit VLAN ID and output port will be sent back to the OFS as an action that instructs the OpenFlow switch to forward a packet. In our example, the packet from VM1 to VM4 will be sent to OFSC, and the VLAN tag will be translated from 100 to 2782. Then the packet is sent to OFSB following a similar process (i.e., translate the VLAN 2782 to 2781), and the packet is sent to Site B by translating 2781 to 200. The gateway system manipulates the two tables to conduct VLAN tag translation between the network domains. With the support of NetFPGA [35], the management of VLAN tags translation is maintained in the control plane and leaves the data-plane processing and tag-translation processing in the line rate.

V. Performance Evaluation

We have deployed the proposed system in operational networks entrusting our design and work in production networks. We implemented the gateway switch based on the NetFPGA programmable switch platform that has four Gigabit Ethernet ports, a Xilinx Virtex-II FPGA, and 4 MB of SRAM. We implemented the controller on a standard Linux PC (2.50GHz Intel Xeon CPU with a 4GB RAM and 1GE network interface). Due to space limitations and the scope of this paper, we focus on the performance results regarding the SDN approach itself in the following discussion. We leave the performance data regarding the achieved throughput of forwarding, the overhead introduced by the proposed mechanisms, and benefits of flexible virtual network deployment to our other publications [3, 32].

A. *Flexibility*

The major goal of the proposed system is to provide network control that is flexible enough to meet a wide range of needs in a heterogeneous environment interconnected by datacenters. Thus, it is important to verify the ability of our system to flexibly specify and instantiate a variety of virtual networks in the cloud comprising multiple datacenters. With the proposed system in this paper, we can easily set up virtual networks as shown in Figure 3 for proof of concept and performance measurement in real production networks. There are three groups of physical machines; the first two (Sites A and C in Figure 3) are located in KUAS, and the third (Site B) is located at another university (NCKU) with eight routing hops between the two universities. The first two groups are allocated in different datacenters and assigned different IP subnets. The traffic between these groups is sent across the WAN environment. All machines use the 2.50GHz Intel Xeon CPU (E5420) with a 4GB RAM and 1GE network interface.

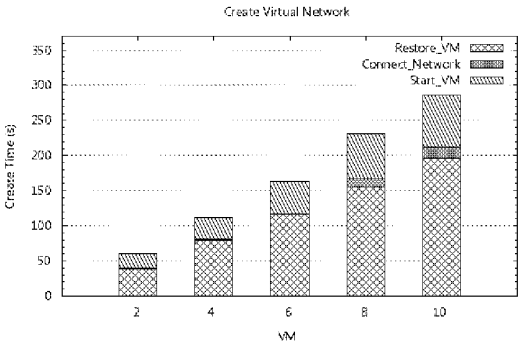


Figure 4: Performance Result of Creation of Virtual Networks

With our system support, we can create virtual networks across three datacenters in a single command. Figure 4 shows the completion time of five virtual networks (2, 4, 6, 8, and 10 nodes in each virtual network, respectively). The completion time is the time from when the network requirement is specified and submitted to when the whole connectivity of this network is established and the traffic can flow successfully (first ping can be responded by each node). The time required in creation of virtual network is only the part labeled with *connect_netwok* in the Figure 4. We can see that our system can deploy virtual networks efficiently. The largest creation time of a virtual network is less than three seconds, in contrast to the previous approaches proposed in [28] in which 27 seconds were needed using VPN.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0037

B. *Scalability*

The size of flow tables in control framework is an important factor for system scalability [13], so we estimate the size of the tables needed by controllers. To emulate a larger environment, we wrote a simulation program based on topology and empirical statistics from our deployment and other publications [15,19,20,21]. In this simulation, we assume a network environment consisting of four datacenters. We perform our analysis by varying the size (number of servers) of each datacenter and the ratio of the cross-site traffic. For example, in Experiment 4 of Figure 5, each datacenter has 50K VMs, and 20% of these VMs have cross-site connections to the nodes in other datacenter. Our simulation is a multithread program, and each thread undergoes the following steps in each experiment: (1) randomly select a VM of one particular datacenter, (2) determine the size of the virtual network based on statistical distribution from system logs on real deployment, (3) randomly select the VMs to make the virtual network and meet its topology requirement, (4) construct the tables required in the controller according the choices made in the previous three steps, and (5) maintain the network for a period time based on the traffic characteristic, then dissolve the network and remove related entries in the tables. The numbers and the required thread are determined by the numbers of hosts and the ratio specified in each experiment.

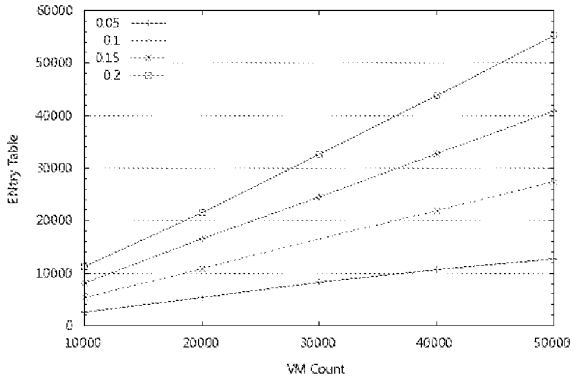


Figure 5: Performance Result of Latency

We realize that other datacenters may have different topologies and the reference data characteristics may not be comprehensive, but our goal here is to provide an initial estimate of our design to cope with large-scale deployment across multiple datacenters. We recorded the maximum number of entries in the controller's table for each experiment in Figure 5. In the largest topology considered in our simulation (Experiment 4), we had a maximum numbers of table entries (55K entries). This initial result shows that the table size is not prohibitively large for a mid-sized cloud with multiple datacenters. A single controller could comfortably handle 10K flow requests per second with such a table size. Please note that not every new flow needs to query the controller; only the new flow whose destination is another datacenter needs to be sent to the controller. However, we observe that the network latency of flow setup time is a significant factor. We perform another measurement in our testbed with some real applications to verify this issue. In the first configuration of this measurement, there is only one controller (located in NCKU, see the Figure 1). As a result, the gateway switch must connect the controller across WAN for flow setup. In the second configuration, we set up another controller in KUAS, so that a portion of flow queries can be handled by the second controller locally. The distributions of response time for flow setup are shown in Figure 6, which is taken in samples over the course of 7 days. The data show that the multiple controllers can improve the flow setup time significantly.

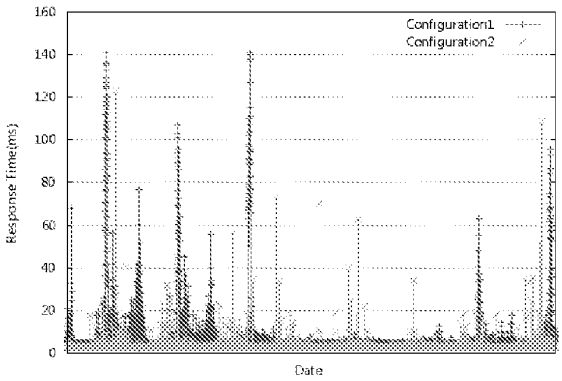


Figure 6: Comparison of Flow Setup Time in Two Configurations

VI. CONCLUSION

We have demonstrated how to leverage the software-defined network concepts and technologies to design and implement an integrated system to address some networking challenges in cloud computing. While the networking mechanism in cloud system is still rapidly evolving, the programmability and extensibility of an SDN-based architecture provide a promising solution to solve pressing problems in cloud networking. Our experience and initial performance results also reveal some important issues for further research. First, although the idea of virtualtransit is promising to solve the interconnection issues among datacenter federation, further investigation are needed on the issues about policy negotiation, security mechanism, and path selection algorithm. Second, more research efforts should be devoted to distributed system issues involved in the design of a control plane scalable to a large network deployment.

ACKNOWLEDGEMENTS

This research was financially supported by the National Science Council of Taiwan, under grants No. NSC 100-2218-E-151-003-MY3 and NSC 101-2622-E-151-026-CC3. The authors also thank NCKU and NCHC for their support on network facilities and WAN configurations.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0038

REFERENCES

[1] C. J. S. Decusatis, A. Carranza, and C. M. Decusatis, "Communication within clouds: open standards and proprietary protocols for data center networking," Communications Magazine, IEEE, vol. 50, pp. 26-33, 2012.

[2] M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Z. Qi, and M. F. Zhani, "Data Center Network Virtualization: A Survey," Communications Surveys & Tutorials, IEEE, vol. 15, pp. 909-928, 2013.

[3] M.-Y. Luo and J.-Y. Chen, "Towards Network Virtualization Management for Federated Cloud Systems", IEEE 6th International Conference on Cloud Computing, June 27-July 2, 2013.

[4] M.-Y. Luo and S.-W. Lin, "From Monolithic Systems to a Federated E-Learning Cloud System," in IEEE International Conference on Cloud Engineering, San Francisco, California, USA, 2013.

[5] J. Mambretti, J. Chen, F. Yeh, T.-L. Liu, M.-Y. Luo, C.-S. Yang, R. v. d. Pol, A. Barczyk, F. Dijkstra, and G. v. Malensteinz, "OpenFlow Services for Science: An International Experimental Research Network Demonstrating Multi-Domain Automatic Network Topology Discovery, Direct Dynamic Path Provisioning Using Edge Signaling and Control, Integration With Multipathing Using MPTCP," in SCinet Research Sandbox at International Conference for high performance Computing, Networking, Storage, and Analysis, 2012.

[6] M.-Y. Luo, J. Chen, J. Mambretti, F. Yeh, and C.-S. Yang, "Network Virtualization Implementation Over Global Research Production Networks," Journal of Internet Technology, 2014.

[7] GENI: Global Environment for Network Innovations, http://www.geni.net

[8] L. Peterson, T. Anderson, D. Blumenthal, D. Casey, D. Clark, D. Estrin, J. Evans, D. Raychaudhuri, M. Reiter, J. Rexford, S. Shenker, and J. Wroclawski, "GENI Design Principles," Computer, vol. 39, pp. 102-105, 2006.

[9] FIRE: Future Internet Research and Experimentation, http://cordis.europa.eu/fp7/ict/fire/

[10] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4D approach to network control and management," ACM SIGCOMM Computer Communication Review, vol. 35, pp. 41-54, 2005.

[11] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: towards an operating system for networks," ACM SIGCOMM Computer Communication Review, vol. 38, pp. 105-110, 2008.

[12] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, pp. 69-74, 2008.

[13] A. Tavakoli, M. Casado, T. Koponen, and S. Shenker, "Applying NOX to the Datacenter," in HotNets, 2009.

[14] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," in ACM SIGCOMM Computer Communication Review, 2007, pp. 1-12.

[15] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, and M. Zhu, "B4: Experience with a globally-deployed software defined WAN," in Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, pp. 3-14, 2013.

[16] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in ACM SIGCOMM Computer Communication Review, 2008, pp. 63-74.

[17] C. Clos, "A study of non-blocking switching networks," Bell System Technical Journal, vol. 32, pp. 406-424, 1953.

[18] C. Kim, M. Caesar, and J. Rexford, "Floodless in seattle: a scalable ethernet architecture for large enterprises," in ACM SIGCOMM Computer Communication Review, pp. 3-14, 2008.

[19] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "PortLand: a scalable fault-tolerant layer 2 data center network fabric," in ACM SIGCOMM Computer Communication Review, pp. 39-50, 2009.

[20] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: a scalable and flexible data center network," in ACM SIGCOMM Computer Communication Review, pp. 51-62, 2009.

[21] J. Mudigonda, P. Yalagandula, J. Mogul, B. Stiekes, and Y. Pouffary, "NetLord: a scalable multi-tenant network architecture for virtualized datacenters," SIGCOMM-Computer Communication Review, vol. 41, p. 62, 2011.

[22] J. Mudigonda, P. Yalagandula, M. Al-Fares, and J. C. Mogul, "SPAIN: COTS Data-Center Ethernet for Multipathing over Arbitrary Topologies," in USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp. 265-280, 2010.

[23] M. Lasserre and V. Kompella, "Virtual private LAN service (VPLS) using label distribution protocol (LDP) signaling," RFC 4762, January 2007.

[24] C. Kim and J. Rexford, "Revisiting Ethernet: Plug-and-play made scalable and efficient," in Local & Metropolitan Area Networks, 2007. LANMAN 2007. 15th IEEE Workshop on, 2007, pp. 163-169.

[25] Andy C. Bavier, Nick Feamster, Mark Huang, Larry L. Peterson and Jennifer Rexford. In VINI Veritas: Realistic and Controlled Network Experimentation. In Proceedings of the SIGCOMM Conference, pages 3-14, Piza, Italy, September 2006.

[26] Xuxian Jiang and Dongyan Xu. Violin: Virtual internetworking on overlay infrastructure. In International Symposium on Parallel and Distributed Processing and Applications (ISPA) 2004, 2004.

[27] S. Bhatia, M. Motiwala, W. Muhlbauer, Y. Mundada, V. Valancius, A. Bavier, N. Feamster, L. Peterson, and J. Rexford. "Trellis: A platform for building flexible, fast virtual networks on commodity hardware". Proc. Workshop on Real Overlays and Distributed Systems, 2008.

[28] T. Wood, K. K. Ramakrishnan, P. Shenoy, and J. Van der Merwe, "CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines," in ACM SIGPLAN Notices, pp. 121-132, 2011.

[29] POX controller. http://www.noxrepo.org/pox/

[30] Beacon: http://beaconcontroller.net/.

[31] Floodlight OpenFlow Controller. http://floodlight.openflowhub.org/.

[32] M.-Y. Luo and J.-Y. Chen, "VirtualTransits: a Platform for Network Virtualization across Data Centers," 2nd IEEE International Conference on Cloud Networking, November 2013.

[33] J. Chase, I. Constandache, A. Demberel, L. Grit, V. Marupadi, M. Sayler, and A. Yumerefendi, "Controlling dynamic guests in a virtual computing utility," In International Conference on the Virtual Computing Initiative (ICVCI 2008), May 2008.

[34] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina. GenericRouting Encapsulation (GRE). Internet Engineering Task Force, March 2000. RFC 2784.

[35] NetFPGA, http://netfpga.org/.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0039

Log In My Cisco
Products & Services Support How to Buy Training & Events Partners

Software-Defined Networks and OpenFlow - The Internet Protocol Journal, Volume 16, No. 1

HOME
ABOUT CISCO
PRESS
THE INTERNET PROTOCOL JOURNAL
ISSUES
VOLUME 16, NUMBER 1, MARCH 2013
From the Editor
SDN and OpenFlow
Address Authentication
WCIT Report
Letters to the Editor
Book Review
Fragments
Call for Papers

by William Stallings

A network organizing technique that has come to recent prominence is the *Software-Defined Network* (SDN) [1]. In essence, an SDN separates the data and control functions of networking devices, such as routers, packet switches, and LAN switches, with a well-defined *Application Programming Interface* (API) between the two. In contrast, in most large enterprise networks, routers and other network devices encompass both data and control functions, making it difficult to adjust the network infrastructure and operation to large-scale addition of end systems, virtual machines, and virtual networks. In this article we examine the characteristics of an SDN, and then describe the *OpenFlow* specification, which is becoming the standard way of implementing an SDN.

Evolving Network Requirements

Before looking in more detail at SDNs, let us examine the evolving network requirements that lead to a demand for a flexible, response approach to controlling traffic flows within a network or the Internet.

One key leading factor is the increasingly widespread use of *Server Virtualization*. In essence, server virtualization masks server resources, including the number and identity of individual physical servers, processors, and operating systems, from server users. This masking makes it possible to partition a single machine into multiple, independent servers, conserving hardware resources. It also makes it possible to migrate a server quickly from one machine to another for load balancing or for dynamic switchover in the case of machine failure. Server virtualization has become a central element in dealing with "big data" applications and in implementing cloud computing infrastructures. But it creates problems with traditional network architectures (for example, refer to [2]). One problem is configuring *Virtual LANs* (VLANs). Network managers need to make sure the VLAN used by the *Virtual Machine* is assigned to the same switch port as the physical server running the virtual machine. But with the virtual machine being movable, it is necessary to reconfigure the VLAN every time that a virtual server is moved. In general terms, to match the flexibility of server virtualization, the network manager needs to be able to dynamically add, drop, and change network resources and profiles. This process is difficult to do with conventional network switches, in which the control logic for each switch is co-located with the switching logic.

Another effect of server virtualization is that traffic flows differ substantially from the traditional client-server model. Typically, there is a considerable amount of traffic among virtual servers, for such purposes as maintaining consistent images of the database and invoking security functions such as access control. These server-to-server flows change in location and intensity over time, demanding a flexible approach to managing network resources.

Another factor leading to the need for rapid response in allocating network resources is the increasing use by employees of mobile devices such as smartphones, tablets, and notebooks to access enterprise resources. Network managers must be able to respond to rapidly changing resource, *Quality of Service* (QoS), and security requirements.

Existing network infrastructures can respond to changing requirements for the management of traffic flows, providing differentiated QoS levels and security levels for individual flows, but the process can be very time-consuming if the enterprise network is large and/or involves network devices from multiple vendors. The network manager must configure each vendor's equipment separately, and adjust performance and security parameters on a per-session, per-application basis. In a large enterprise, every time a new virtual machine is brought up, it can take hours or even days for network managers to do the necessary reconfiguration [3].

This state of affairs has been compared to the mainframe era of computing [4]. In the era of the mainframe, applications, the operating system, and the hardware were vertically integrated and provided by a single vendor. All of these ingredients were proprietary and closed, leading to slow innovation. Today, most computer platforms use the x86 instruction set, and a variety of operating systems (Windows, Linux, or Mac OS) run on top of the hardware. The OS provides APIs that enable outside providers to develop applications, leading to rapid innovation and deployment. In a similar fashion, commercial networking devices have proprietary features and specialized control planes and hardware, all vertically integrated on the switch. As will be seen, the SDN architecture and the OpenFlow standard provide an open architecture in which control functions are separated from the network device and placed in accessible control servers. This setup enables the underlying infrastructure to be abstracted for applications and network services, enabling the network to be treated as a logical entity.

SDN Architecture

Figure 1 illustrates the logical structure of an SDN. A central controller performs all complex functions, including routing, naming, policy declaration, and security checks. This plane constitutes the *SDN Control Plane*, and consists of one or more SDN servers.

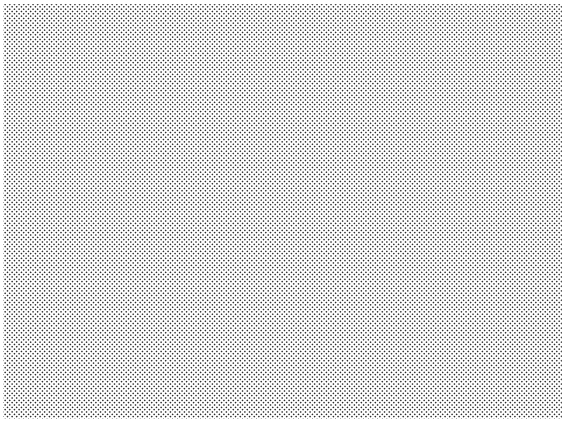


Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0040

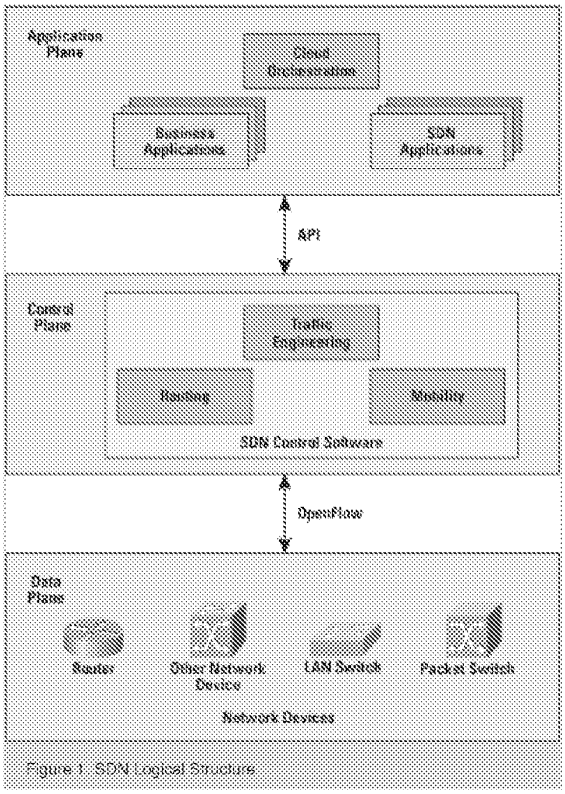


Figure 1: SDN Logical Structure

The *SDN Controller* defines the data flows that occur in the *SDN Data Plane*. Each flow through the network must first get permission from the controller, which verifies that the communication is permissible by the network policy. If the controller allows a flow, it computes a route for the flow to take, and adds an entry for that flow in each of the switches along the path. With all complex functions subsumed by the controller, switches simply manage flow tables whose entries can be populated only by the controller. Communication between the controller and the switches uses a standardized protocol and API. Most commonly this interface is the OpenFlow specification, discussed subsequently.

The SDN architecture is remarkably flexible: it can operate with different types of switches and at different protocol layers. SDN controllers and switches can be implemented for Ethernet switches (Layer 2), Internet routers (Layer 3), transport (Layer 4) switching, or application layer switching and routing. SDN relies on the common functions found on networking devices, which essentially involve forwarding packets based on some form of flow definition.

In an SDN architecture, a switch performs the following functions:

- The switch encapsulates and forwards the first packet of a flow to an SDN controller, enabling the controller to decide whether the flow should be added to the switch flow table.
- The switch forwards incoming packets out the appropriate port based on the flow table. The flow table may include priority information dictated by the controller.
- The switch can drop packets on a particular flow, temporarily or permanently, as dictated by the controller. Packet dropping can be used for security purposes, curbing Denial-of-Service (DoS) attacks or traffic management requirements.

In simple terms, the SDN controller manages the forwarding state of the switches in the SDN. This management is done through a vendor-neutral API that allows the controller to address a wide variety of operator requirements without changing any of the lower-level aspects of the network, including topology.

With the decoupling of the control and data planes, SDN enables applications to deal with a single abstracted network device without concern for the details of how the device operates. Network applications see a single API to the controller. Thus it is possible to quickly create and deploy new applications to orchestrate network traffic flow to meet specific enterprise requirements for performance or security.

SDN Domains

In a large enterprise network, the deployment of a single controller to manage all network devices would prove unwieldy or undesirable. A more likely scenario is that the operator of a large enterprise or carrier network divides the whole network into numerous nonoverlapping SDN domains as shown in Figure 2.

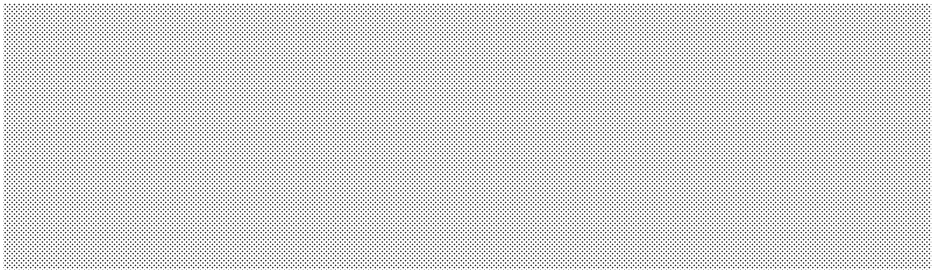


Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0041

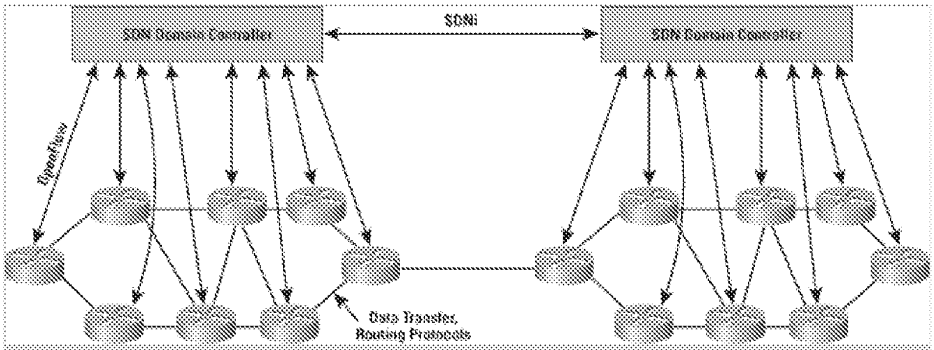


Figure 2: SDN Domain Structure

Reasons for using SDN domains include the following:

- *Scalability:* The number of devices an SDN controller can feasibly manage is limited. Thus, a reasonably large network may need to deploy multiple SDN controllers.
- *Privacy:* A carrier may choose to implement different privacy policies in different SDN domains. For example, an SDN domain may be dedicated to a set of customers who implement their own highly customized privacy policies, requiring that some networking information in this domain (for example, network topology) not be disclosed to an external entity.
- *Incremental deployment:* A carrier's network may consist of portions of traditional and newer infrastructure. Dividing the network into multiple, individually manageable SDN domains allows for flexible incremental deployment.

The existence of multiple domains creates a requirement for individual controllers to communicate with each other via a standardized protocol to exchange routing information. The IETF is currently working on developing a protocol, called *SDNi,* for "interfacing SDN Domain Controllers" [5]. SDNi functions include:

- Coordinate flow setup originated by applications containing information such as path requirement, QoS, and service-level agreements across multiple SDN domains.
- Exchange reachability information to facilitate inter-SDN routing. This information exchange will allow a single flow to traverse multiple SDNs and have each controller select the most appropriate path when multiple such paths are available.

The message types for SDNi tentatively include the following:

- Reachability update
- Flow setup/tear-down/update request (including application capability requirements such as QoS, data rate, latency etc.)
- Capability update (including network-related capabilities such as data rate and QoS, and system and software capabilities available inside the domain)

OpenFlow

To turn the concept of SND into practical implementation, two requirements must be met. First, there must be a common logical architecture in all switches, routers, and other network devices to be managed by an SDN controller. This logical architecture may be implemented in different ways on different vendor equipment and in different types of network devices, so long as the SDN controller sees a uniform logical switch function. Second, a standard, secure protocol is needed between the SDN controller and the network device.

Both of these requirements are addressed by *OpenFlow,* which is both a protocol between SDN controllers and network devices, as well as a specification of the logical structure of the network switch functions [6, 7]. OpenFlow is defined in the *OpenFlow Switch Specification,* published by the *Open Networking Foundation* (ONF). ONF is a consortium of software providers, content delivery networks, and networking equipment vendors whose purpose is to promote software-defined networking.

This discussion is based on the current OpenFlow specification, Version 1.3.0, June 25, 2012 [8]. The original specification, 1.0, was developed at Stanford University and was widely implemented. OpenFlow 1.2 was the first release from ONF after inheriting the project from Stanford. OpenFlow 1.3 significantly expands the functions of the specification. Version 1.3 is likely to become the stable base upon which future commercial implementations for OpenFlow will be built. ONF intends for this version to be a stable target for chip and software vendors, so little if any change is planned for the foreseeable future [9].

Logical Switch Architecture

Figure 3 illustrates the basic structure of the OpenFlow environment. An SDN controller communicates with OpenFlow-compatible switches using the OpenFlow protocol running over the *Secure Sockets Layer* (SSL). Each switch connects to other OpenFlow switches and, possibly, to end-user devices that are the sources and destinations of packet flows. Within each switch, a series of tables—typically implemented in hardware or firmware—are used to manage the flows of packets through the switch.

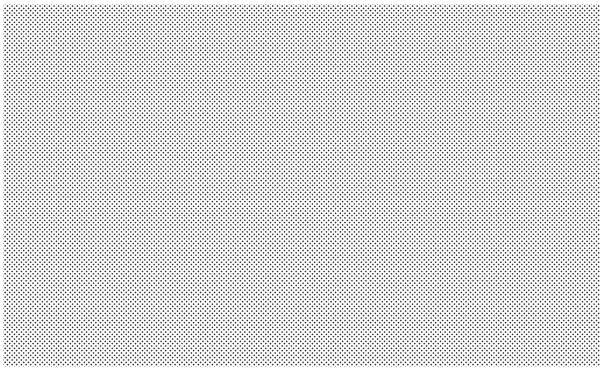


Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0042

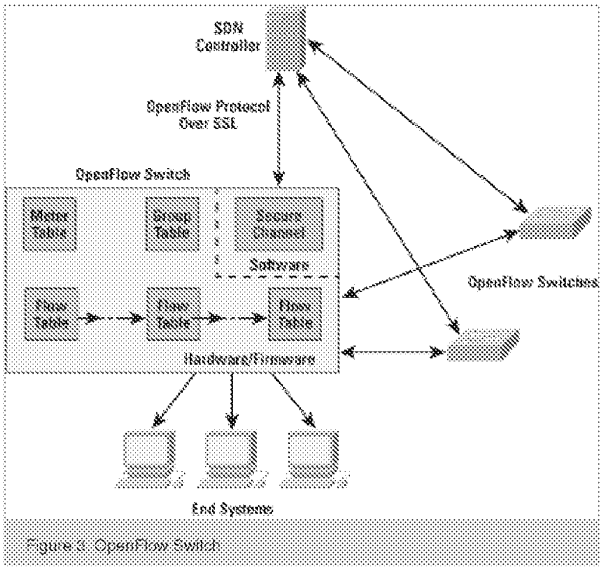


Figure 3: OpenFlow Switch

The OpenFlow specification defines three types of tables in the logical switch architecture. A *Flow Table* matches incoming packets to a particular flow and specifies the functions that are to be performed on the packets. There may be multiple flow tables that operate in a pipeline fashion, as explained subsequently. A flow table may direct a flow to a *Group Table,* which may trigger a variety of actions that affect one or more flows. A *Meter Table* can trigger a variety of performance-related actions on a flow.

Before proceeding, it is helpful to define what the term flow means. Curiously, this term is not defined in the OpenFlow specification, nor is there an attempt to define it in virtually all of the literature on OpenFlow. In general terms, a flow is a sequence of packets traversing a network that share a set of header field values. For example, a flow could consist of all packets with the same source and destination IP addresses, or all packets with the same VLAN identifier. We provide a more specific definition subsequently.

Flow Table Components

The basic building block of the logical switch architecture is the flow table. Each packet that enters a switch passes through one or more flow tables. Each flow table contains entries consisting of six components:

- *Match Fields:* Used to select packets that match the values in the fields.
- *Priority:* Relative priority of table entries.
- *Counters:* Updated for matching packets. The OpenFlow specification defines a variety of timers. Examples include the number of received bytes and packets per port, per flow table, and per flow-table entry; number of dropped packets; and duration of a flow.
- *Instructions:* Actions to be taken if a match occurs.
- *Timeouts:* Maximum amount of idle time before a flow is expired by the switch.
- *Cookie:* Opaque data value chosen by the controller. May be used by the controller to filter flow statistics, flow modification, and flow deletion; not used when processing packets.

A flow table may include a table-miss flow entry, which renders all Match Fields wildcards (every field is a match regardless of value) and has the lowest priority (priority 0). The Match Fields component of a table entry consists of the following required fields:

- *Ingress Port:* The identifier of the port on the switch where the packet arrived. It may be a physical port or a switch-defined virtual port.
- *Ethernet Source and Destination Addresses:* Each entry can be an exact address, a bitmasked value for which only some of the address bits are checked, or a wildcard value (match any value).
- *IPv4 or IPv6 Protocol Number:* A protocol number value, indicating the next header in the packet.
- *IPv4 or IPv6 Source Address and Destination Address:* Each entry can be an exact address, a bitmasked value, a subnet mask value, or a wildcard value.
- *TCP Source and Destination Ports:* Exact match or wildcard value.
- *User Datagram Protocol (UDP) Source and Destination Ports:* Exact match or wildcard value.

The preceding match fields must be supported by any OpenFlow-compliant switch. The following fields may be optionally supported:

- *Physical Port*: Used to designate underlying physical port when packet is received on a logical port.
- *Metadata:* Additional information that can be passed from one table to another during the processing of a packet. Its use is discussed subsequently.
- *Ethernet Type:* Ethernet Type field.
- *VLAN ID and VLAN User Priority:* Fields in the IEEE 802.1Q Virtual LAN header.
- *IPv4 or IPv6 DS and ECN:* Differentiated Services and Explicit Congestion Notification fields.
- *Stream Control Transmission Protocol (SCTP) Source and Destination Ports:* Exact match or wildcard value.
- *Internet Control Message Protocol (ICMP) Type and Code Fields:* Exact match or wildcard value.
- *Address Resolution Protocol (ARP) Opcode:* Exact match in Ether-net Type field.
- *Source and Target IPv4 Addresses in Address Resolution Protocol (ARP) Payload:* Can be an exact address, a bitmasked value, a subnet mask value, or a wildcard value.
- *IPv6 Flow Label:* Exact match or wildcard.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0043

- *ICMPv6 Type and Code fields:* Exact match or wildcard value.
- *IPv6 Neighbor Discovery Target Address:* In an IPv6 Neighbor Discovery message.
- *IPv6 Neighbor Discovery Source and Target Addresses:* Link-layer address options in an IPv6 Neighbor Discovery message.
- *Multiprotocol Label Switching (MPLS) Label Value, Traffic Class, and Bottom of Stack (BoS):* Fields in the top label of an MPLS label stack.

Thus, OpenFlow can be used with network traffic involving a variety of protocols and network services. Note that at the MAC/link layer, only Ethernet is supported. Thus, OpenFlow as currently defined cannot control Layer 2 traffic over wireless networks.

We can now offer a more precise definition of the term *flow*. From the point of view of an individual switch, a flow is a sequence of packets that matches a specific entry in a flow table. The definition is packet-oriented, in the sense that it is a function of the values of header fields of the packets that constitute the flow, and not a function of the path they follow through the network. A combination of flow entries on multiple switches defines a flow that is bound to a specific path.

The *instructions component* of a table entry consists of a set of instructions that are executed if the packet matches the entry. Before describing the types of instructions, we need to define the terms "Action" and "Action Set." Actions describe packet forwarding, packet modification, and group table processing operations. The OpenFlow specification includes the following actions:

- *Output:* Forward packet to specified port.
- *Set-Queue:* Sets the queue ID for a packet. When the packet is forwarded to a port using the output action, the queue id determines which queue attached to this port is used for scheduling and forwarding the packet. Forwarding behavior is dictated by the configuration of the queue and is used to provide basic QoS support.
- *Group:* Process packet through specified group.
- *Push-Tag/Pop-Tag:* Push or pop a tag field for a VLAN or MPLS packet.
- *Set-Field:* The various Set-Field actions are identified by their field type; they modify the values of respective header fields in the packet.
- *Change-TTL:* The various Change-TTL actions modify the values of the IPv4 Time To Live (TTL), IPv6 Hop Limit, or MPLS TTL in the packet.

An *Action Set* is a list of actions associated with a packet that are accumulated while the packet is processed by each table and executed when the packet exits the processing pipeline. Instructions are of four types:

- *Direct packet through pipeline:* The Goto-Table instruction directs the packet to a table farther along in the pipeline. The Meter instruction directs the packet to a specified meter.
- *Perform action on packet:* Actions may be performed on the packet when it is matched to a table entry.
- *Update action set:* Merge specified actions into the current action set for this packet on this flow, or clear all the actions in the action set.
- *Update metadata:* A metadata value can be associated with a packet. It is used to carry information from one table to the next.

Flow-Table Pipeline

A switch includes one or more flow tables. If there is more than one flow table, they are organized as a pipeline as shown in Figure 4, with the tables labeled with increasing numbers starting with 0.

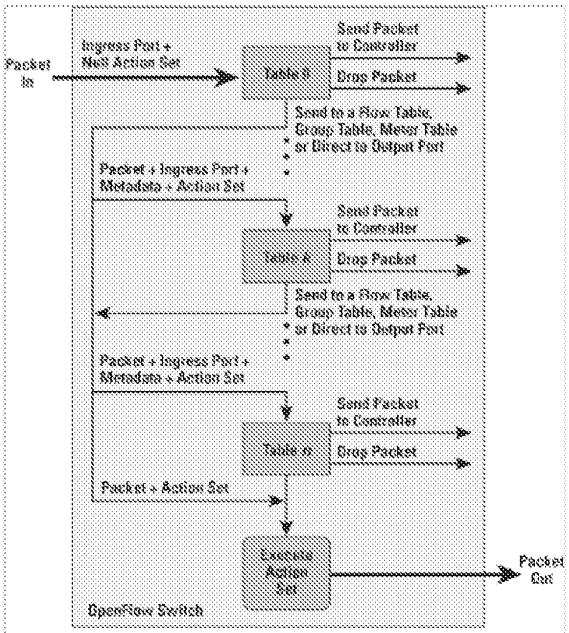


Figure 4: Packet Flow Through OpenFlow Compliant Switch

When a packet is presented to a table for matching, the input consists of the packet, the identity of the ingress port, the associated metadata value, and the associated action set. For Table 0, the metadata value is blank and the action set is null. Processing proceeds as follows:

1. Find the highest-priority matching flow entry. If there is no match on any entry and there is no table-miss entry, then the packet is

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0044

dropped. If there is a match only on a table-miss entry, then that entry specifies one of three actions:

a. Send packet to controller. This action will enable the controller to define a new flow for this and similar packets, or decide to drop the packet.

b. Direct packet to another flow table farther down the pipeline.

c. Drop the packet.

2. If there is a match on one or more entries other than the table-miss entry, then the match is defined to be with the highest-priority matching entry. The following actions may then be performed:

a. Update any counters associated with this entry.

b. Execute any instructions associated with this entry. These instructions may include updating the action set, updating the metadata value, and performing actions.

c. The packet is then forwarded to a flow table further down the pipeline, to the group table, or to the meter table, or it could be directed to an output port.

For the final table in the pipeline, forwarding to another flow table is not an option.

If and when a packet is finally directed to an output port, the accumulated action set is executed and then the packet is queued for output.

OpenFlow Protocol

The OpenFlow protocol describes message exchanges that take place between an OpenFlow controller and an OpenFlow switch. Typically, the protocol is implemented on top of SSL or Transport Layer Security (TLS), providing a secure OpenFlow channel.

The OpenFlow protocol enables the controller to perform add, update, and delete actions to the flow entries in the flow tables. It supports three types of messages, as shown in Table 1.

- *Controller-to-Switch:* These messages are initiated by the controller and, in some cases, require a response from the switch. This class of messages enables the controller to manage the logical state of the switch, including its configuration and details of flow- and group-table entries. Also included in this class is the Packet-out message. This message is used when a switch sends a packet to the controller and the controller decides not to drop the packet but to direct it to a switch output port.
- *Asynchronous:* These types of messages are sent without solicitation from the controller. This class includes various status messages to the controller. Also included is the Packet-in message, which may be used by the switch to send a packet to the controller when there is no flow-table match.
- *Symmetric:* These messages are sent without solicitation from either the controller or the switch. They are simple yet helpful. Hello messages are typically sent back and forth between the controller and switch when the connection is first established. Echo request and reply messages can be used by either the switch or controller to measure the latency or bandwidth of a controller-switch connection or just verify that the device is operating. The Experimenter message is used to stage features to be built into future versions of OpenFlow.

Table 1: OpenFlow Messages

Name	Description
Controller-to-Switch	
Features	Request the capabilities of a switch. Switch responds with a features reply that specifies its capabilities.
Configuration	Set and query configuration parameters. Switch responds with parameter settings.
Modify-State	Add, delete, and modify flow/group entries and set switch port properties.
Read-State	Collect information from switch, such as current configuration, statistics, and capabilities.
Packet-out	Direct packet to a specified port on the switch.
Barrier	Barrier request/reply messages are used by the controller to ensure message dependencies have been met or to receive notifications for completed operations.
Role-Request	Set or query role of the OpenFlow channel. Useful when switch connects to multiple controllers.
Asynchronous-Configuration	Set filter on asynchronous messages or query that filter. Useful when switch connects to multiple controllers.
Asynchronous	
Packet-in	Transfer packet to controller.
Flow-Removed	Inform the controller about the removal of a flow entry from a flow table.
Port-Status	Inform the controller of a change on a port.
Error	Notify controller of error or problem condition.
Symmetric	
Hello	Exchanged between the switch and controller upon connection startup.
Echo	Echo request/reply messages can be sent from either the switch or the controller, and they must return an echo reply.
Experimenter	For additional functions.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0045

The OpenFlow protocol enables the controller to manage the logical structure of a switch, without regard to the details of how the switch implements the OpenFlow logical architecture.

Summary

SDNs, implemented using OpenFlow, provide a powerful, vendor-independent approach to managing complex networks with dynamic demands. The software-defined network can continue to use many of the useful network technologies already in place, such as virtual LANs and an MPLS infrastructure. SDNs and OpenFlow are likely to become commonplace in large carrier networks, cloud infrastructures, and other networks that support the use of big data.

References

[1] Greg Goth, "Software-Defined Networking Could Shake Up More than Packets," *IEEE Internet Computing,* July/August, 2011.

[2] Robin Layland, "The Dark Side of Server Virtualization," *Network World,* July 7, 2010.

[3] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," ONF White Paper, April 12, 2012.

[4] Dell, Inc., "Software Defined Networking: A Dell Point of View," Dell White Paper, October 2012.

[5] Steven Vaughan-Nichols, "OpenFlow: The Next Generation of the Network?" *Computer,* August 2011.

[6] Thomas A. Limoncelli, "OpenFlow: A Radical New Idea in Networking," *Communications of the ACM,* August 2012.

[7] Open Networking Foundation, "OpenFlow Switch Specification Version 1.3.0," June 25, 2012.

[8] Sean Michael Kerner, "OpenFlow Protocol 1.3.0 Approved," *Enterprise Networking Planet*, May 17, 2012.

WILLIAM STALLINGS is an independent consultant and author of many books on security, computer networking, and computer architecture. His latest book is *Data and Computer Communications* (Pearson, 2013). He maintains a computer science resource site for computer science students and professionals at ComputerScienceStudent.com. He has a Ph.D. in computer science from M.I.T. He can be reached at ws@shore.net

Information For
Small Business
Midsize Business
Service Provider
Industries
Marketplace
Contacts
Contact Cisco
Meet our Partners
Find a Reseller

News & Alerts
Newsroom
Blogs
Field Notices
Security Advisories
Technology Trends
Cloud
Internet of Things (IoT)
Software Defined Networking (SDN)

Support
Downloads
Documentation
Communities
DevNet
Learning Network
Support Community
Video Portal

About Cisco
Investor Relations
Corporate Social Responsibility
Environmental Sustainability
Trust and Transparency Center
There's Never Been A Better Time
Careers
Search Jobs
We Are Cisco
Programs
Cisco Designated VIP Program
Cisco Powered
Financing Options

Contacts | Feedback | Help | Site Map | Terms & Conditions | Privacy Statement | Cookie Policy | Trademarks

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0046

OpenRoads: Empowering Research in Mobile Networks

Kok-Kiong Yap, Masayoshi Kobayashi, Rob Sherwood, Nikhil Handigol,
Te-Yuan Huang, Michael Chan, and Nick McKeown
Stanford University, NEC System Platforms Research Labs, and Deutsche Telekom Inc., R&D Lab
yapkke@stanford.edu, m-kobayashi@eo.jp.nec.com, robert.sherwood@telekom.com,
{nikhilh,huangty,mcfchan,nickm}@stanford.edu

1. INTRODUCTION

Many researchers are working on mobility management. Continued innovation and development in this area can naturally be expected. We anticipate rapid progress will be made in the areas of handover between different technologies, separation of "virtual" wireless service providers from the physical infrastructure, lossless handovers and multi-homing solutions. There is no lack of good ideas. However, testing and verifying these ideas at a significant scale has proven to be difficult. The community will therefore benefit from a common platform, in which they can reuse and build upon each other's work. To this end, we developed *OpenRoads*, a platform for innovation in mobile networks.

The goal of OpenRoads is to create an *open* platform in which other researchers and ourselves can experiment and explore different mobility solutions, network controllers and/or routing protocols. OpenRoads provides researchers control of the network through two means, namely, control of the datapath using OpenFlow [2, 3] and control of the device configuration using SNMP.

Via control given by OpenFlow and SNMP, we further provide abstract events and functions for network management in OpenRoads. The abstraction in turns allows us to manage with ease heterogeneous network devices, namely different wireless technologies such as WiFi and WiMAX. Further, sharing the vision of a virtualized network, such as in [1, 5], OpenRoads also presents our attempt to share network control among different researchers.

Through our poster, shown in Fig. 1, we present OpenRoads to the community at large. Our long term goal is to develop OpenRoads into a robust and effective platform for research in mobile networks, that is adopted by other campuses, for experimentation in their production networks.

2. CONTROLLING A NETWORK SLICE

A major distinction between today's cellular network and IP-based network is the control plane. IP has the control plane embedded in each and every router, resulting in an inflexible infrastructure with glacially slow innovation. On the other hand, today's cellular network separates the control plane and rapid innovations in the control plane are common.

A typical Ethernet switch consists of the flow table in hardware and the control logic in software, all of which is enclosed in a single device. Adopting a similar philosophy as the cellular network, OpenFlow separates the control logic of a switch from the physical device, exporting it to an external controller. By controlling the flow table of a switch from an external controller, OpenFlow allows rapid innovation to the datapath of the network. To deploy OpenFlow in our closet (and closets of other researchers), we are adding OpenFlow as feature to commercial switches and wireless devices, such as WiFi APs and WiMAX base-station.

Building on OpenFlow, we developed a platform OpenRoads, which allows researchers to control the network via OpenFlow and SNMP. Our resulting architecture is as shown in right half of Fig. 1. While OpenFlow provides control of forwarding, SNMP provides control of device configuration. Being a widely deployed interface, SNMP allows researchers to configure the switches and wireless access points in a network. Configuration is especially critical in wireless devices, where parameters like transmit power can directly impact on the datapath performance. Further, controllers can also be promptly informed of network events such as a host joining a WiFi AP.

To further share this control of the network among multiple researchers, we use the FlowVisor [4] and SNMP demultiplexer to flexibly divide the control. This is achieved through appropriate proxying, rewriting and policing of the control messages. More specifically, each experiment is given a "flow-space" (a range of header values). By ensuring that experiments cannot exert control beyond their designated "flow-space", FlowVisor provides isolation between experiments. Also, by appearing as a controller to network devices and as devices to the controllers, FlowVisor (and SNMP demultiplexer) allows researchers to develop as if they have complete over the entire network. This allows the scale of the experiment to be expanded without any change in code.

Finally, to allow for rapid development on OpenRoads, we provide abstract high-level control interfaces, such as redirection of a flow in the network and retrieving all flows associated with a host. These interfaces can be categorized as communication (i.e., messaging between devices), flow management (e.g., install flow entries for specific route), and device control (e.g., set power level of WiFi APs).

3. PROGRESS

OpenFlow has been ported as a feature in many switches from multiple vendors, including NEC and HP. Many of these devices already supports SNMP. We have also extended OpenFlow and SNMP into WiFi APs built with PCEngine boxes running Linux, and a NEC WiMAX basestation. Armed with these devices, we have deployed a wireless access network consisting of 5 switches, 30 WiFi APs and a WIMAX basestation. With this physical infrastructure, we are able to validate the usefulness of OpenRoads.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0047

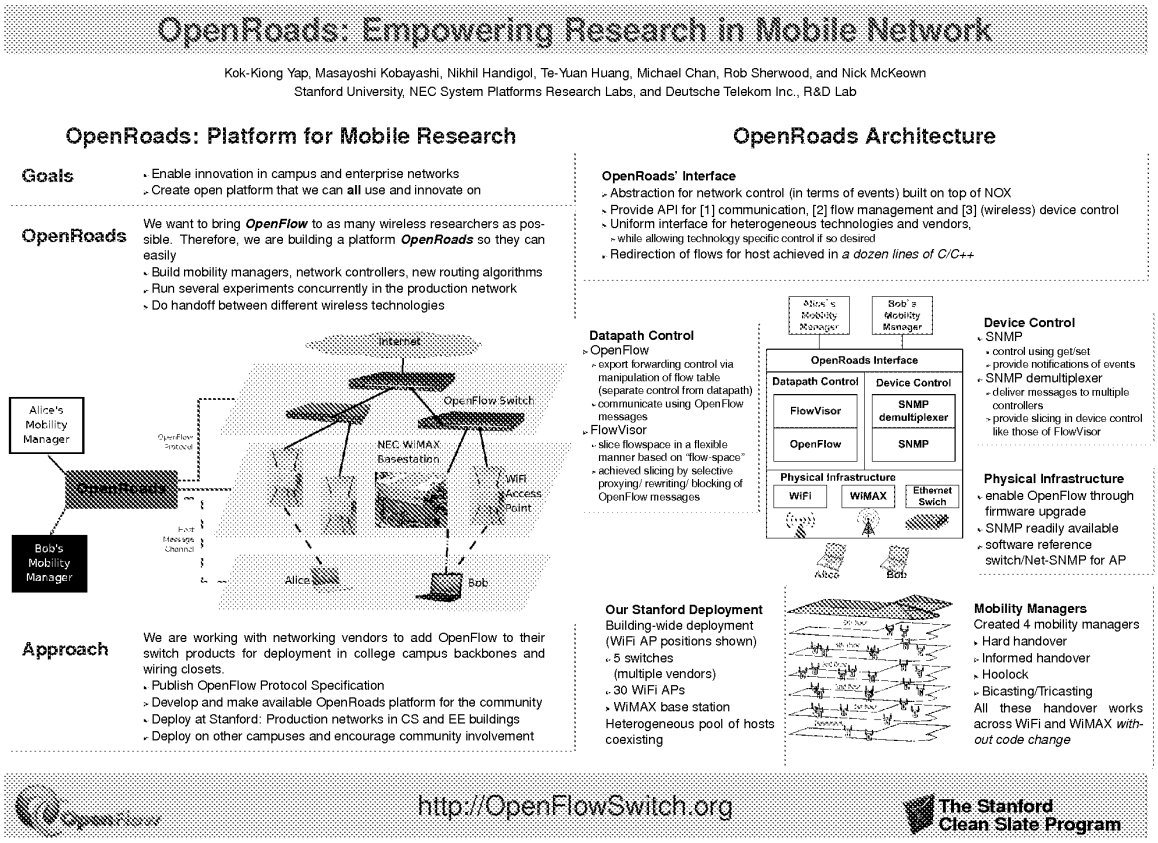


Figure 1: OpenRoads' Poster

Using our OpenRoads platform, we were able to create multiple mobility managers (some of which were created as part of a class) and run them concurrently on a single physical infrastructure. The mobility managers includes hard handover, informed handover, bicasting/tricasting and Hoolock. In informed handover, the client is provided with information to aid handover decisions. While in bicasting/tricasting, flows are multicasted to the client over multiple interfaces. Hoolock is a novel handover scheme which also exploits multiple interfaces on a device to achieve lossless handover, without the need to reorder at the client. All of the above mobility managers are able to perform handover between WiFi and WiMAX without any code change. In these mobility managers, OpenRoads' interface allows us to redirect flows associated with a host in 12 lines of C/C++.

While running these mobility managers, we have a heterogeneous pool of hosts coexisting in the network. For example, the bicasting mobility assumes all devices have more than one network interfaces, while hard handover assumes the contrary, i.e., all devices has one interface. These devices coexists in our network, and are handled by their respective mobility managers without conflict.

We have created a network in which multiple experiments can be run concurrently over the same physical infrastructure. In the long run, we hope OpenRoads will inspire the creation of virtual service providers that are independent of the infrastructure, allowing for seamless handover between different wireless technologies.

4. REFERENCES

[1] N. Feamster, L. Gao, and J. Rexford. How to lease the internet in your spare time. *SIGCOMM Comput. Commun. Rev.*, 37(1):61–64, 2007.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, April 2008.

[3] The OpenFlow Switch Consortium. http://www.openflowswitch.org.

[4] Rob Sherwood, Michael Chan,Glen Gibb, Nikhil Handigol,Te-Yuan Huang,Peyman Kazemian,Masayoshi Kobayashi, David Underhill, Kok-Kiong Yap, and Nick McKeown. Carving research slices out of your production networks with OpenFlow. In *Proceedings of ACM SIGCOMM (Demo)*, Barcelona, Spain, August 2009.

[5] A virtual network infrastructure. http://www.vini-veritas.net.

OpenRadio: A Programmable Wireless Dataplane

Manu Bansal, Jeffrey Mehlman, Sachin Katti, Philip Levis
Stanford University
{manub, jmehlman, skatti}@stanford.edu, pal@cs.stanford.edu

Abstract

We present OpenRadio, a novel design for a programmable wireless dataplane that provides modular and declarative programming interfaces across the entire wireless stack. Our key conceptual contribution is a principled refactoring of wireless protocols into processing and decision planes. The processing plane includes directed graphs of algorithmic actions (eg. 54Mbps OFDM WiFi or special encoding for video). The decision plane contains the logic which dictates which directed graph is used for a particular packet (eg. picking between data and video graphs). The decoupling provides a declarative interface to program the platform while hiding all underlying complexity of execution. An operator only expresses decision plane rules and corresponding processing plane action graphs to assemble a protocol. The scoped interface allows us to build a dataplane that arguably provides the right tradeoff between performance and flexibility. Our current system is capable of realizing modern wireless protocols (WiFi, LTE) on off-the-shelf DSP chips while providing flexibility to modify the PHY and MAC layers to implement protocol optimizations.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Design, Management, Performance

Keywords

Programmable, Infrastructure, Dataplane, Wireless

1. INTRODUCTION

Wireless protocols evolve continuously. For example, 3GPP (the standards body responsible for specifying the LTE cellular standard) expects to release major updates to the basic LTE standard once every 18 months, and minor updates continuously. Even within a deployed operational network, operators and vendors need to continuously optimize the network to handle problems such as inter-cell interference, new traffic classes such as video and so on. For example, operators need to implement management mechanisms that dynamically adjust spectrum and power allocation at neighboring basestations at a fine-grained subcarrier level on timescales of hundreds of milliseconds to ensure that mobile handsets at the edge of both cell-sites are not adversely affected. Similarly, to cope with the growing popularity of video over spectrum starved cellular networks, operators would prefer to use a PHY layer optimization such as unequal error protection (UEP) - where certain frames (I-frames) are better protected than others (P- or B-frames) [15][25] - that can deliver much higher video quality while using lesser spectral resources. Similar protocol changes and optimizations abound (some more examples are discussed in Sec. 2) that are beneficial to implement for a variety of reasons.

The wireless infrastructure is expected to support such evolvability. In the past, such changes typically required replacing the basestations, since the protocol definition was closely coupled with the hardware in the form of specific ASICs designed for each protocol. Such an approach worked fine when protocol upgrades were relatively slow (several years) and basestations were deployed sparsely and covered large areas. However, in current networks, protocol changes are continuous (once every few months) and networks are dense (e.g. one basestation per block in urban areas) to provide high capacity and uniform coverage. Physically and frequently upgrading the basestations for such dense networks is prohibitively expensive.

In this paper, we argue that wireless infrastructure needs a programmable dataplane to support such evolvability. Specifically, the basestations should be remotely programmable to enable vendors and operators to upgrade and optimize the network completely in software. To build such functionality, we present OpenRadio, a novel design for a *programmable wireless network dataplane*. Built around a commodity multi-core hardware platform, the core component of OpenRadio is a software abstraction layer that exposes a modular and declarative interface to program the PHY (baseband) and MAC layers. An operator describes a protocol by defining *rules* to match subsets of traffic streams and specifying *actions* to process them. Rules are logical predicates on parameters of packets such as header fields, received signal strength, channel frequency and other fields that may be programmed. Actions describe behavior such as encoding/decoding of data and scheduling of traffic on the channel.

OpenRadio's design makes two contributions. First, it decouples wireless protocol definition from the hardware, even while ensuring that commodity multi-core platforms can be used for implementing the protocols. Such decoupling is feasible because we can identify low-level atomic building blocks in wireless protocols at the PHY

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.
HotSDN'12, August 13, 2012, Helsinki, Finland.
Copyright 2012 ACM 978-1-4503-1477-0/12/08 ...$15.00.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0049

and MAC layers that are shared across different protocols (different versions of LTE 4G, 3G, WiFi and so on). Further, OpenRadio shows that commodity multi-core platforms are capable of executing such atomic blocks efficiently. Consequently, defining and customizing protocol definition programmatically is feasible, as long as modular abstractions exist to re-program individual protocol elements.

OpenRadio's second contribution is the design of this software abstraction layer that exposes a modular and declarative interface to program wireless protocols. The key idea is a principled decomposition of wireless protocols into separate *processing* and *decision* plane components with a simple, well-understood API between them. This decomposition is the main enabler of an efficient implementation of a flexible and modular wireless programming interface. The processing plane includes *actions*: operations that only involve data-manipulation on packet data but no branching logic. The decision plane includes *rules*: all decision logic that picks sequences of operations to transform data stream subsets. Such decomposition lends itself to efficient real-time implementation on commodity digital signal processors (DSPs) that are widely available from a number of chip vendors [26, 12, 9, 14].

OpenRadio can be used to specify both the underlying protocols as well as optimizations. For example, a rule for WiFi would require all received packets matching the 6Mbps data rate in the PHY header to be processed with the action of BPSK demodulation. Protocol optimizations can be expressed by adding rules and actions to an existing definition. For video delivery, an operator using OpenRadio might optimize for this class of traffic by inserting a new rule to match all video packets (based on a header field). The rule would require them to be processed through a special action of UEP (en)decoding. The UEP (en)decoder action would also be programmed by the operator as an atomic block. Meanwhile, the existing standard WiFi-data handling remains untouched.

More generally, our longer-term goal is to design a programmable wireless dataplane that includes the basestations, backhaul network switches/routers and gateways to realize a software defined cellular infrastructure. Our goal in this paper, however, is more modest. It is to discuss the design of a programmable basestation that forms one (albeit important) component of this infrastructure. At this stage, we have a prototype implementation of OpenRadio that supports the WiFi PHY on off-the-shelf multicore DSP processors from TI [26]. The platform is capable of fully processing the WiFi PHY protocol in software running at the highest speeds (upto 108Mbps) and meeting all protocol deadlines. We pick WiFi for proof-of-concept since its processing requirements are the most stringent due to wide-bandwidth, multi-carrier modulation and random-access deadlines on the order of microseconds. We plan to add support for LTE in the near future, as well as leverage the dataplane to start researching the design of a control plane for cellular SDN infrastructure.

In the following section, we discuss more specific examples of the kinds of protocol changes that operators would want to implement in their networks remotely, and that OpenRadio's design supports.

2. USE CASES

Programmable basestations for cellular infrastructure are being increasingly favored over fixed-function hardware as exemplified by offerings of "software-upgradable" platforms for HSPA/WCDMA/LTE from some of the top vendors [23, 3]. Still, many desirable features either remain hard to program or are simply not possible to implement. A well-designed programmable dataplane can enable these features. Let us consider some specific use-cases to gain insight into the kind of programmability required.

2.1 Cell-size based optimization

OFDMA leads to power-amplifier (PA) inefficiencies due to a high peak-to-average-power ratio (PAPR) as compared to single-carrier modulation. This lead to the choice of SC-FDMA for LTE (Rel 8/9) uplink to save user equipment (UE) power. However, advances in PA technology have now led to the adoption of OFDMA (Rel 10) in favor of higher spectral efficiency. This is especially true for small-cells where the UE needs to transmit at a lower power than a macro-cell. The switch from SC-FDMA to OFDMA effectively involves toggling a DFT block on the PHY signal chain. Definitions of individual processing blocks remain unchanged.

Another cell-size based optimization could involve picking the right FFT size to balance multipath effects against PAPR inefficiencies. A higher FFT size leads to better multipath resiliency but a higher PAPR value which is less energy efficient. A smaller cell can use a smaller FFT size for the same data delivery performance since it suffers less multipath fading and delay spread. Similar FFT size adaptation based on link length can also benefit WiFi. The FFT processing block only needs to be configurable in FFT size without any changes in the algorithm.

2.2 Co-existence of heterogeneous cells

Cellular networks are rapidly becoming heterogeneous in cell size with deployment of small-cells. Femto-cells enhance coverage in a small area such as a home but generally provide access to a closed group of subscribers. However, deployment in the same frequency as the containing macro-cell can cause strong interference to external users in the same region. This has led to techniques like Almost Blank Subframes (ABS) [6] that co-ordinate the use of time-frequency resource elements between the two cells to avoid interference. Such co-ordination can be implemented with programmability in the MAC scheduler that decides whether or not to use a resource element at any given time.

2.3 Application-specific wireless service

A promising feature of LTE is multiple service profiles [4] with varying throughput and latency characteristics. The intent of the QoS profiles is to provide customized network service to diverse applications. However, the standard only specifies profile parameters while leaving the mechanisms open to implementation. A programmable dataplane provides the vehicle necessary to implement various profiles and allowing their extension to new application classes. Different processing chains can be tied to different traffic classes. Prioritization, scheduling of traffic and error recovery mechanisms can also be programmed according to application needs. Those features can also evolve with new applications such as public safety networks, urban sensing and media broadcast in addition to classic voice, video and data delivery. A programmable dataplane can similarly facilitate implementation of QoS features for WiFi, WiMAX, White-space networking etc.

2.4 Evolving standards

Cellular wireless standards have been upgrading every couple of years [1, 2]. Users are ready to consume network upgrades at an even faster pace with an average UE lifespan of 18 months [10]. Yet, standards take a decade or more to reach their peak market share [11]. Arguably, the biggest impediment to evolution of the infrastructure is its inability to be programmed in-place and incrementally.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0050

Wireless networks are already operating with very high spectral efficiency (within $\approx 20\%$ of capacity limits). Consequently, signal processing motifs are stabilizing with little room for improvement. Moreover, cellular networks are designed such that most intelligence is place in the infrastructure while keeping UEs simple. This allows a bigger playing fields on the base-station side that is still interoperable with existing users. Most gains are to be had by optimizing networks according to the operational scenario and load patterns, which tend to be very diverse [11]. At the same time, the set of users is always a rich mix of capabilities as new standards take long to penetrate. The ideal base-station must be able to cater to all of them simultaneously. OpenRadio provides a platform where new protocols and features can be "plugged-in" seamlessly. Its programmable dataplane provides flexibility in selection, configuration and arrangement of data processing motifs to compose protocols optimized to the operational scenario.

3. DESIGN GOALS

Our goal is to design a programmable wireless dataplane with a modular interface that can provide both the performance and flexibility necessary for implementing a wide class of modern wireless protocols and their optimizations.

As a *performance* benchmark, the platform must be capable of processing a bandwidth of at least 20MHz for OFDM-complexity protocols, including but not limited to IEEE 802.11 WiFi, IEEE 802.16 WiMAX and 3GPP LTE-Advanced. The platform must provide enough computation throughput for these applications as well as the processing latency requirements associated with specific protocols. In the case of WiFi, the most challenging deadline is an ACK-turnaround time of $25\mu s$ for every successfully decoded data packet.

Flexibility is a more qualitative characteristic than performance. Our design goal is to provision the system to allow a smooth tradeoff between performance and deviation from the standard configuration. Small variations in behavior should be enabled with little or no performance loss. For example, a version of WiFi optimized for video delivery using UEP only requires a change in the encoding of I- and B-/P-frame bits into a packet of the PHY layer. All other operations are preserved. Consequently, the platform must be able to deliver near-WiFi computation performance for video-WiFi. At the same time, the platform should not preclude more significant changes from being realized at all.

An equally important design goal is making the OpenRadio platform easy to program. The system must provide *modularity* such that the involvement of a programmer is limited to only the affected components of the protocol. Adapting a WiFi implementation to video-WiFi should not require the operator to modify or even understand the OFDM (de)modulator or the CRC implementation. Without such modularity, the utility of a flexible dataplane will be far from practical.

4. PROGRAMMING ABSTRACTIONS

The OpenRadio dataplane design provides a network substrate that can be programmed using a modular, declarative interface, provides high performance capable of supporting the highest rate modern wireless protocols, and can be built using commodity DSP silicon. In this section, we describe its declarative interface.

4.1 Modular declarative interface

OpenRadio's dataplane enables the network operator to declaratively program the network stack for specific traffic subsets. He only specifies what packet processing behavior he expects from the network for different traffic classes. However, the operator does not need to specify how the packet processing behavior is integrated into an operational network and realized in practice. This low-level complexity is abstracted out by the interface.

OpenRadio's declarative interface is built on a *principled refactoring of decision and processing paths in wireless protocols*. Specifically, we observe that protocols can be finely partitioned into two separate parts: processing blocks which specify how a protocol transforms analog waveforms into bits, and decision logic which specifies when different processing transformations are used. For example, the 54Mbps OFDM decoding sequence of 802.11g is a processing specification, while interpreting the signal field of the 802.11 PHY layer control header to determine that the payload will be at 54Mbps is a decision specification.

The processing plane encapsulates the series of signal processing algorithms used in a PHY processing chain where each module corresponds to a single algorithm. For example, the processing plane will have modules for FFT, 64-QAM mapping and slicing, convolutional encoding, Viterbi decoding, etc. These modules correspond to *blocks* in a signal processing chain. The decision plane encapsulates all the decision logic functionality. It is responsible for selecting processing plane blocks to be used for encoding or decoding a packet. For example, the decision plane logic for 802.11g will appropriately configure and stitch the processing plane plane blocks to decode a 54Mbps packet without needing to touch any block internals. Similarly, a protocol modification that rearranges the same signal processing blocks would only need to modify the decision plane, leaving the processing plane blocks untouched. As another example, LTE changes the way packets are processed on the downlink and uplink. It uses standard multicarrier OFDM for the downlink, while it uses a modified single carrier modulation (SC-FDM) for uplink to conserve mobile client batteries [5]. The details of the signal processing algorithms are unimportant, but the main take-away is that both OFDM and SC-FDM are simply different ways of configuring and stitching basic signal processing blocks such as FFTs, channel decoders and slicers. The decision plane can simply specify such re-configurations and re-arrangements to realize different protocol optimizations.

The two key choices - granularity of processing plane blocks and decoupling decision plane logic from the processing plane - are directly motivated by the goal of modular programmability. Decoupling allows logical changes and rearrangement of signal processing blocks to be limited to the decision plane without requiring any understanding of the processing plane. Reconfiguring the behavior of a processing plane block limits changes to the specific block without affecting other blocks or the decision plane. Adding a new block (eg. a new channel encoder or decoder optimized for video) is similarly a contained effort. The level of abstraction of processing plane blocks (eg. FFT vs. low level instructions such as complex multiply and accumulate) is high enough to allow most programmability to be expressed declaratively and allows convenient reuse of code. The imperative part is abstracted out in processing plane blocks. Such separation of processing and decision planes allows an intuitive programming model made up of rules and actions.

4.2 Rule-action programming model

The decision plane of a protocol has a natural representation as a set of *rules* encoding the decision logic. These rules govern the processing plane by naming *actions* that must be performed depending on the outcome of rule evaluation. For example, a rule-action pair will read `if hdr.rate == 54Mbps, perform action_decode_54Mbps`.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0051

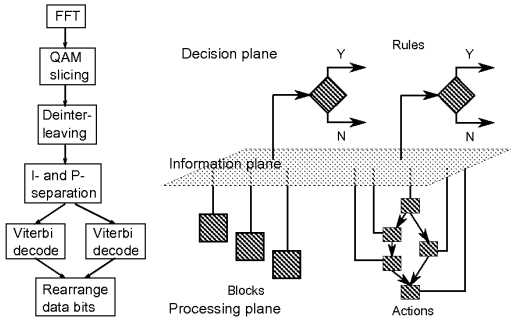


Figure 1: (a) Directed acyclic graph of an action customized for video delivery. (b) OpenRadio programming model depicting the separation of decision and processing planes and their relation with the information plane.

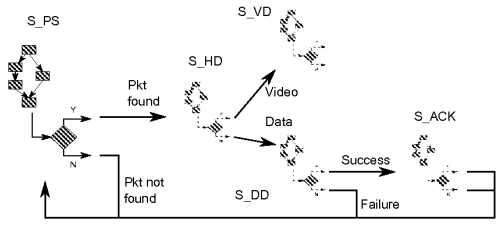


Figure 2: A state machine with states for packet search (S_PS), header decode (S_HD), video decode (S_VD), data decode (S_DD) and sending an acknowledgement (S_ACK).

Actions are directed acyclic graphs (DAGs) of signal-processing operations. Each operation is typically simple: it takes values from an input stream, applies a simple operation such as an FFT, and outputs the processed stream. Operations do not include conditionals or other complex, dynamic logic since that is factored out into the decision plane. An example action DAG for a video decoding chain is shown in fig. 1(a).

The regular interface of a processing plane block is composed of the data inputs (eg. complex samples) and data output (eg. soft bits, complex samples). In addition, a block has configuration inputs (eg. FFT size) and output statistics (eg. energy value for a packet-detect block, rate field for a header-parsing block). Collection of all configurations and statistics from the processing plane forms an *information plane* that acts as the bi-directional interface between decision and processing planes (fig. 1(b)). Rules are written on elements of the information plane. Similarly, the decision plane exercises control over the processing plane through configuration primitives made available on the information plane.

The programming model of rules and actions is a specific concrete representation of the concept of decision and processing plane decoupling. It lets us express protocols declaratively by preserving the decoupling semantics. Actions only specify the functional behavior and data dependencies without specifying any execution schedules, hardware placement details or resource allocation. Similarly, rules only specify the logical content without dictating how they will be implemented. Thus, the model provides a generic API without being tied to any particular system design or hardware platform architecture. It also allows us to leverage extensive prior work where the stream computation in actions is naturally expressed using dataflow languages [16]. The current implementation of OpenRadio uses annotated C code, but its decoupled programming model makes it simple to integrate new domain-specific dataflow programming languages.

4.3 State machine model and deadlines

Rules and actions are assembled into a state machine to express a fully-functional protocol, as in fig. 2. Each state specifies exactly one action to be performed and contains rules for transitioning to other states. For example, the header decode state will perform the action of decoding the header field and include rules to transition to data-decoding states, one for each data rate. Such a state machine model matches naturally with the specification of protocols, some of which use formal state machine languages in the standard itself.

In addition to rules and actions, a first class component of wireless protocols is deadlines. Deadlines may apply to specific actions such as a maximum of $1\mu s$ delay in assessing channel idle/busy state for effective channel backoff implementation in WiFi (i.e. slot-time stipulation), or to a series of actions such as a maximum delay of $25\mu s$ in decoding a packet from the end-of-arrival (due to WiFi ACK-turnaround stipulation.) In the latter case, they exist between certain states, and consist of hard timing requirements on all of the computation (decision and processing-plane) that occurs in the interval between those two states. Valid deadlines may exist between any two states in a protocol but there must be a logical route through the rules from the initial state to the end state.

Similar to the choice of rule-action model, the state machine model is a specific choice for representing protocols that is intuitive and preserves the declarative nature of the programming interface including specification of deadlines. Further, it allows us to leverage prior work on formal state machine languages like [7] with strong semantics to facilitate static analysis that is critical to providing guaranteed execution times and meeting deadlines as we discuss further in sec. 5.

4.4 Challenges

The key challenge in OpenRadio is efficient realization of the decoupled declarative programming model. In particular, the system must be able to meet protocol deadlines reliably without requiring the programmer to optimize implementation specifics. We can summarize the challenges as follows:

- Efficient execution of computationally demanding processing plane actions ($\approx$100 Gflops) while retaining hardware abstraction, programming modularity and flexibility.
- Maintaining realtime property over a sequence of states, i.e., under serial composition, in order to meet deadlines reliably.

The rich programming interface made up of rules-actions and state-machine models is the key feature that distinguishes OpenRadio from a bare-bones software-radio platform or a programmable substrate like an FPGA platform that may offer similar performance. Let us discuss the principles behind an efficient implementation of this programming interface.

5. DESIGN AND IMPLEMENTATION

Decoupling of decision and processing planes allows us to provide a modular and declarative programming interface through the rule, action and state machine models. Can we design a system that implements such an interface with the level of computational performance needed for modern protocols with 20-80MHz of bandwidth while meeting tight latency deadlines on the order of tens of microseconds? We answer the question in the affirmative in this

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0052

Processing plane block	WiFi	LTE	3G	DVB-T
FIR/IIR	✓	✓	✓	✓
Correlation	✓	✓	✓	✓
Spreading			✓	
FFT	✓	✓		✓
Channel Estimation	✓	✓	✓	✓
QAM Mapping	✓	✓	✓	✓
Interleaving	✓	✓	✓	✓
Convolutional Coding	✓	✓	✓	✓
Turbo Coding		✓	✓	
Randomization	✓	✓	✓	✓
CRC	✓	✓	✓	

Table 1: Many computationally intensive processing plane blocks are common to all modern PHYs, differing only in configuration.

section describing the design and early results demonstrating its feasibility. Our system is based on two keys principles.

5.1 Design principles

Learning from existing programmable solutions, OpenRadio follows *the principle of scoping flexibility*. We aim to provide enough flexibility to implement most protocols and optimizations of interest but no more. For example, the FFT block only needs to provide configurable FFT-length, not a choice of different Fourier Transform algorithms. Similarly, the convolutional decoder needs to be configurable in the coding rate (say, 1/2, 1/3, 1/4, 1/5) or constraint length over small parameter ranges but the actual algorithm (Viterbi or BCJR) is immaterial. There is, however, still scope for blocks that need almost arbitrary flexibility such as a bit interleaver.

Restricting the scope of flexibility where advantageous allows for optimizing the hardware to perform some of the relatively-fixed operations extremely efficiently while being able to provide the requisite configurability. The merit of this approach lies in the observation that many of the heavy computational processing plane blocks of modern PHY protocols are common to most standards and fall in this category (Table 1). Further, commodity DSP platforms with hybrid architectures optimizing common *compute kernels* are becoming widely available, either through loosely-coupled on-chip accelerators as in TI KeyStone [26] or through tightly-coupled special-purpose instructions as in CEVA-XC Family [9], [20][14][12]. A well-designed accelerator can provide speed-ups of up to an order of magnitude (10-25x) over pure-software [13]. As a price reference point, a TI KeyStone chip housing tens of accelerators is marked at less than $200 and is capable of processing 80MHz WiFi-like intensity of computation, i.e. it easily supports WiFi, WiMAX, LTE and W-CDMA.

While the above hardware substrate provides the raw horsepower needed for a programmable dataplane, OpenRadio's dataplane must provide a modular declarative interface while ensuring that packets are processed within the deadlines of the protocol. To this end, OpenRadio follows *the key principle of decision and processing plane separation throughout its system design*. In contrast with providing modularity and declarative interfaces, the motivation here is to achieve near-deterministic execution times for processing plane actions. On a hybrid architecture, blocks implemented through hardware accelerators would finish in predictable amounts of time by default. This may not be true of other software-based blocks that will run on CPUs or DSPs. The algorithmic reason for such unpredictability is presence of branches in arbitrary code. For example, a soft-slicer block that could do any of BPSK, QPSK, 16-QAM or 64-QAM, will run for different amounts of time in each mode. However, a branch is precisely a decision plane rule. The factoring of algorithm and code into decision and processing planes pushes out all such branches to the decision plane, leaving only branch-free code in processing plane blocks.

Combining the decoupling principle with assisting implementation choices, we not only achieve deterministic-runtime blocks but also preserve the property under composition as required for actions. This is the key property that allows OpenRadio to systematically meet execution deadlines. An added advantage of this principle is a direct mapping of declarative programs to execution primitives which greatly simplifies the system design.

5.2 Preliminary implementation

The design of OpenRadio system is focused towards utilizing multi-core DSP architectures. They typically contain DSP cores optimized for signal processing computation and hardware accelerators providing speed-up of specific, commonly used algorithms. Such platforms provide the most desired tradeoff between performance and flexibility. Our specific hardware choice is the TI KeyStone architecture [26]. The software challenge is to harness the raw compute horsepower while retaining modular and declarative abstractions.

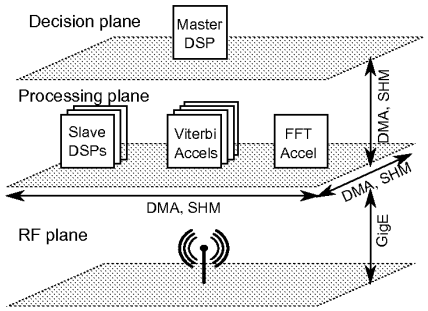


Figure 3: OpenRadio basestation internals with different kinds of compute cores and an RF frontend.

The core objective of the runtime system is *fast and deterministic execution of actions*. While efficient execution is achieved through accelerated hardware, determinism in runtime is obtained through decision/processing separation down to the software level, mimicking the programming model itself. The system architecture is shown in fig. 3. The same separation is applied to hardware resources as well, designating the decision plane core as the master core controlling the processing plane slave cores and accelerators.

The decision plane runtime emulates the protocol state-machine by evaluating rules as transitions are made between states. The action associated with each state is computed on the processing plane by assigning constituent blocks to slave cores (or accelerators) according to a pre-computed execution and resource allocation schedule. Such static scheduling is made possible because of our specific implementation choices such as software-addressed L2 memory and event-polling instead of interrupts. They eliminate major sources of indeterminism from the underlying hardware architecture as well. Further, using a message-passing model for inter-processor communication allows accurate accounting of IPC latencies. Full description of implementation and scheduling framework are beyond the scope of this paper. However, it is worth stating that the scheduling of actions in our deterministic model bears strong resemblance with instruction loop-pipelining in compilers. This reduction of model allows us to leverage compiler scheduling algorithms as in [24].

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0053

Our early prototype is capable of processing full 54Mbps 802.11g PHY layer stack on a single KeyStone chip with more than one full DSP core worth of computer power still available for implementing protocol modifications. The utilization figure includes the overhead introduced by the runtime software that strives to maintain modularity. This overhead measures to roughly 20% of the entire compute load.

5.3 Design limitations

Our software system design applies equally well to other multi-core processors, hybrid or symmetric. Still, it is based on certain judicious assumptions.

First, OpenRadio assumes that the majority of the computation time is spent in the processing plane of the protocol, and decision overhead is negligible. However, heavy decision planes would incur prohibitive inter-core communication overheads implying that decision/processing plane separation is not the best design choice. Similarly, a heavy sequential algorithm on the processing plane would restrict the benefits of a multi-core architecture if it forbids pipelined parallelism.

Second, OpenRadio assumes decision/processing refactoring is meaningful in signal processing blocks. However, some blocks that need to take decisions, like packet-search block, are best expressed with some embedded decision code. If the gap in average and worst-case run times becomes too high, maintaining the deterministic semantic on that block could become very inefficient.

6. RELATED WORK AND DISCUSSION

OpenRadio has a lot in common with SDN efforts such as OpenFlow [18], both in goals and in challenges. The key design consideration is to pick the right abstractions to balance flexibility against performance. However, there are some major differences too. Interoperation with existing devices on the Internet is a major consideration for layer-3 SDN interfaces. For OpenRadio, inter-operation is much less of an issue as it operates on layers 1 and 2 which are contained within autonomous domains. Further, a programmable wireless dataplane has a massive computational load that must be processed under strict deadlines, a challenge unique to wireless.

There are comparable solutions for programmable wireless dataplanes that excel in some dimension at the cost of others. GNURadio [8] offers unlimited flexibility through purely software-based processing and convenient programming through high-level interfaces at the cost of performance and real-time guarantees. SORA [27] offers full flexibility and sufficient performance while arguably sacrificing ease of programming by nearly by-passing the desktop operating system. WARP [17] and AirBlue [22] use an FPGA approach to retain flexibility equal to software while gaining performance but end up driving the price-point high. Others like NI PXIe-1082 SDR [21] have taken an all-out approach with unrestrained FPGA and CPU horsepower resulting in extremely expensive systems (≈$50k per SDR platform) only good for small-scale laboratory research and prototyping. Yet other cellular domain platforms [3][26, 12, 9, 14] tend to provide the right price-performance and flexibility point but remain hard to program. This has led to the development of higher-level interfaces [22, 19] to work with these platforms.

OpenRadio strikes a unique combination of the programmability-performance-price metrics by treating all three as primary objectives from the get-go. The design is based on principled refactoring of wireless protocols into processing and decision planes. It is general enough to be implemented on a variety of platforms including DSPs, FPGAs and multi-core desktop machines. Code has enough structure that it can be statically analyzed to yield guaranteed real-time performance and efficient hardware use while still exposing modular, declarative interfaces.

Our intent is to provide programmability across the entire wireless stack. In this paper, most of our design choices are guided by PHY-layer considerations. The decision and processing plane model applies equally well to the MAC layer - the processing plane operation is to map multiple traffic queues onto the channel frequency-time grid while the decision plane operation is to pick the right mapping according to a programmable algorithm. The weighting of computation between the two planes may be different than that in PHY. This exploration is part of our ongoing research.

7. REFERENCES

[1] http://en.wikipedia.org/wiki/3GPP.

[2] http://en.wikipedia.org/wiki/Comparison_of_mobile_phone_standards.

[3] RBS6000 Base Station. *http://www.ericsson.com/ourportfolio/products/base-stations.*

[4] 3GPP TS 23.203 V8.3.1.

[5] 3GPP TS 36.201 - v1.0.0. LTE Physical Layer - General Description.

[6] Alcatel-Lucent. R1-093340: Blank Subframes for LTE. *3GPP TSG RAN WG1,meeting 58, Shenzhen, China, August 2009.*

[7] M. Anlauff. XASM - An Extensible, Component-Based Abstract State Machines Language. In *International Workshop on Abstract State Machines, Lecture Notes on Computer Science (LNCS)*, pages 69–90. Springer-Verlag, 2000.

[8] E. Blossom. GNU radio: tools for exploring the radio frequency spectrum. *Linux J.*, 2004:4–, June 2004.

[9] CEVA, Inc. CEVA-XC High-Performance, Low-Power DSP Cores for Advanced Wireless Communications.

[10] Earth911.com. Facts about cell phones. *http://earth911.com/recycling/electronics/cell-phone/facts-about-cell-phones/.*

[11] Ericsson Corp. Traffic and Market Report, June 2012. *http://www.ericsson.com/res/docs/2012/traffic_and_market_report_june_2012.pdf.*

[12] Freescale, Inc. StarCore Digital Signal Processors.

[13] R. Hameed, W. Qadeer, M. Wachs, O. Azizi, A. Solomatnikov, B. C. Lee, S. Richardson, C. Kozyrakis, and M. Horowitz. Understanding sources of inefficiency in general-purpose chips. ISCA '10. ACM, 2010.

[14] Intel Corporation, Inc. Intel Solutions for the Next Generation Multi-Radio Basestation. *Application Note.*

[15] ISO/IEC Moving Picture Experts Group (MPEG). The MPEG4 Standard. *ISO/IEC 14496 - Coding of audio-visual objects*, 1998.

[16] W. M. Johnston, J. R. P. Hanna, and R. J. Millar. Advances in dataflow programming languages. *ACM Comput. Surv.*, 36(1):1–34, Mar. 2004.

[17] A. Khattab. WARP: A flexible platform for clean-slate wireless medium access protocol design. *SIGMOBILE Mob. Comp. Comm.*, January 2008.

[18] N. McKeown. OpenFlow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38:69–74, March 2008.

[19] Microsoft Research. Brick Specification. *The SORA Project*, 2011.

[20] Mindspeed Technologies, Inc. MindSpeed Baseband Processors.

[21] National Instruments, Inc. NI PXIe-1082 SDR Chassis. *http://sine.ni.com/nips/cds/view/p/lang/en/nid/207346.*

[22] A. Ng, K. E. Fleming, M. Vutukuru, S. Gross, Arvind, and H. Balakrishnan. Airblue: A System for Cross-Layer Wireless Protocol Development. 2010.

[23] Nokia Siemens Networks. Flexi Multiradio BTS. *http://www.nokiasiemensnetworks.com/portfolio/products/mobile-broadband/single-ran-advanced/flexi-multiradio-base-station.*

[24] J. Sánchez and A. González. The Effectiveness of Loop Unrolling for Modulo Scheduling in Clustered VLIW Architectures. ICPP '00. IEEE Computer Society, 2000.

[25] S. Sen, S. Gilani, S. Srinath, S. Schmitt, and S. Banerjee. Design and Implementation of an "Approximate" Communication System for Wireless Media Applications. In *ACM SIGCOMM*, 2009.

[26] Texas Instruments. TMS320TCI6616 Communications Infrastructure KeyStone SoC. *Data Manual.*

[27] J. Zhang. Experimenting software radio with the SORA platform. In *Proceedings of ACM SIGCOMM*, 2010.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0054

2012 European Workshop on Software Defined Networking

Toward Software-Defined Cellular Networks

Li Erran Li
Bell Labs
erranlli@research.bell-labs.com

Z. Morley Mao
University of Michigan
zmao@umich.edu

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

***Abstract*—Existing cellular networks suffer from inflexible and expensive equipment, complex control-plane protocols, and vendor-specific configuration interfaces. In this position paper, we argue that software defined networking (SDN) can simplify the design and management of cellular data networks, while enabling new services. However, supporting many subscribers, frequent mobility, fine-grained measurement and control, and real-time adaptation introduces new scalability challenges that future SDN architectures should address. As a first step, we propose extensions to controller platforms, switches, and base stations to enable controller applications to (i) express high-level policies based on subscriber attributes, rather than addresses and locations, (ii) apply real-time, fine-grained control through local agents on the switches, (iii) perform deep packet inspection and header compression on packets, and (iv) remotely manage shares of base-station resources.**

I. Introduction

The growing popularity of smart phones and tablet computers places an increasing strain on cellular networks. Yet, despite tremendous innovation in mobile applications, the cellular network infrastructure is remarkably brittle. Software defined networking (SDN) can simplify network management, while enabling new services. However, supporting many subscribers, frequent mobility, fine-grained measurement and control, and real-time adaptation introduces scalability challenges that future SDN architectures should address.

A. Today's LTE Cellular Data Networks

Long Term Evolution (LTE) cellular networks [1] connect base stations (eNodeB) to the Internet using IP networking equipment, as shown in Figure 1. The user equipment (UE) connects to a base station, which directs traffic through a serving gateway (S-GW) over a GPRS Tunneling Protocol (GTP) tunnel. The S-GW serves as a local mobility anchor that enables seamless communication when the user moves from one base station to another. The S-GW must handle frequent changes in a user's location, and store a large amount of state since users retain their IP addresses when they move. The S-GW tunnels traffic to the packet data network gateway (P-GW). The P-GW enforces quality-of-service policies and monitors traffic to perform billing. The P-GW also connects to the Internet and other cellular data networks, and acts as a firewall that blocks unwanted traffic. The policies at the P-GW can be very fine-grained, based on whether the user is roaming, properties of the user equipment, usage caps in the service contract, parental controls, and so on.

Besides data-plane functionalities, the base stations, serving gateways, and packet gateways also participate in several

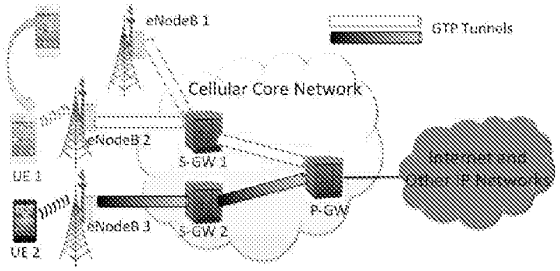


Fig. 1. LTE data plane

control-plane protocols, as illustrated in Figure 2. In coordination with the Mobility Management Entity (MME), they perform hop-by-hop signaling to handle session setup, teardown, and reconfiguration, as well as mobility, e.g., location update, paging, and handoff. For example, in response to a UE's request for dedicated session setup (e.g., for VoIP call), the P-GW sends QoS and other session information (e.g., the TCP/IP 5-tuple) to the S-GW. The S-GW in turn forwards the information to the MME. The MME then asks the base station to allocate radio resources and establish the connection to the UE. During handoff of a UE, the source base station sends the handoff request to the target base station. After receiving an acknowledgement, the source base station transfers the UE state (e.g., buffered packets) to the target base station. The target base station also informs the MME that the UE has changed cells, and the previous base station to release resources (e.g., remove the GTP tunnel).

The S-GW and P-GW are also involved in routing, running protocols such as OSPF. The Policy Control and Charging Function (PCRF) manages flow-based charging in the P-GW. The PCRF also provides the QoS authorization (QoS class identifier and bit rates) that decides how to treat each traffic flow, based on the user's subscription profile. QoS policies can be dynamic, e.g., based on time of day. This must be enforced at the P-GW. The Home Subscriber Server (HSS) contains subscription information for each user, such as the QoS profile, any access restrictions for roaming, and the associated MME. In times of cell congestion, a base station reduces the max rate allowed for subscribers according to their profiles, in coordination with the P-GW.

Today's cellular network architectures have several major limitations. Centralizing data-plane functions such as monitoring, access control, and quality-of-service functionality at the packet gateway introduces scalability challenges. This

problem

978-0-7695-4870-8/12 $26.00 © 2012 IEEE
DOI 10.1109/EWSDN.2012.28

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0055

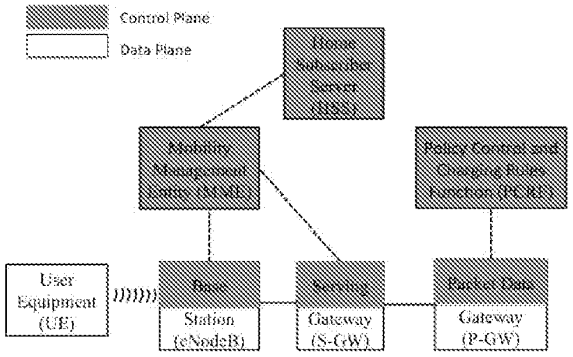


Fig. 2. Simplified LTE network architecture

makes the equipment very expensive (e.g., more than 6 million dollars for a Cisco packet gateway). Centralizing data-plane functions at the cellular-Internet boundary forces all traffic through the P-GW, including traffic between users on the same cellular network, making it difficult to host popular content *inside* the cellular network. In addition, the network equipment has vendor-specific configuration interfaces, and communicate through complex control-plane protocols, with a large and growing number of tunable parameters (e.g., several thousand parameters for base stations). As such, carriers have (at best) indirect control over the operation of their networks, with little ability to create innovative services.

B. SDN for Cellular Data Networks

Cellular data networks are ripe for the introduction of Software Defined Networking (SDN), where the network equipment performs basic packet-processing functions at the behest of applications running on a logically-centralized controller. In the next section, we discuss how SDN could give cellular operators greater control over their equipment, simplify network management, and introduce value-added services. SDN can enable carriers to distribute data-plane rules over multiple, cheaper network switches, reducing the scalability pressure on the packet gateway and enabling flexible handling of traffic that stays within the cellular network.

Supporting real-time updates to many fine-grained packet-handling rules raises significant scalability challenges. Frequent user mobility can require forwarding state at the level of individual subscribers, and the state must change quickly to avoid service disruptions. To detect when subscribers exceed their usage caps, the switches must perform fine-grain monitoring of traffic volumes. The network must adapt quickly to the measurement data to adapt QoS policies, or transcode content to offer good service during times of congestion. To address these challenges, we propose four main extensions to controllers, switches, and base stations:

Flexible policies on subscriber attributes: Many controller applications need to apply policy based on the properties of cellular subscribers, including the network provider, device type, subscriber type, and recent usage. The controller should automatically translate policies based on subscriber attributes to packet-processing rules based on IP addresses and network locations.

Scalability through local switch agents: The need for fast and frequent updates to a large amount of data-plane state would put tremendous pressure on a central controller. Instead, switches should run software agents that perform simple local actions (such as polling traffic counters and comparing against thresholds), at the behest of the controller.

Flexible switch patterns and actions: Today's OpenFlow already supports flexible packet classification and actions. That said, cellular networks would benefit from support for deep-packet inspection, header compression, and message-based control protocols like SCTP [2].

Remote control of virtualized base-station resources: Virtualizing the base station by time slot and subcarriers can give different applications, traffic classes, or virtual operators the illusion of a dedicated base station. An open API between the controller and base station can enable remote control of radio resource allocation, admission control, handoff, and paging.

For prototyping our architecture, we are encouraged by the availability of the open source LTE base station code provided by openairinterface.org and the open source LTE packet core code provided by openepc.net.

II. Controller Applications

In this section, we identify several major challenges in cellular data networks, and how Software Defined Networking can enable better solutions.

A. Directing Traffic Through Middleboxes

Cellular network operators offer many fine-grained services implemented in network appliances, or middleboxes. In a dynamic environment, cellular providers need to adapt video quality in real time based on cell tower congestion, device type, and the subscriber's service plan [3]. To improve security for enterprise customers, providers direct traffic through intrusion detection and prevention systems. Certain legacy devices need in-network support for echo cancellation for VoIP calls. However, today's cellular carriers do not have fine-grained control over routing, forcing them to either direct excess traffic through unnecessary middleboxes or manage an unwieldy set of tunnels.

SDN provides fine-grained packet classifier and flexible routing, which can easily direct a chosen subset of traffic through a set of middleboxes. As a result, middleboxes will handle much less traffic, making them much cheaper. In addition, with support for deep-packet inspection, SDN switches could support some middlebox functionality directly, reducing the number of extra devices in the network.

B. Monitoring for Network Control & Billing

Due to frequent user mobility, and changing channel conditions, cellular networks are subject to rapid changes in traffic demands, and frequent signaling messages to migrate connections. Real-time traffic monitoring is crucial for triggering fast

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0056

adaptation. This includes load balancing data traffic from base stations to a different S-GW, and from a different S-GW to a different P-GW, load balancing control traffic from the base station and S-GW to a different MME. Real-time monitoring also enables rapid per-application content adaptation (e.g., video conferencing, or streaming from Netflix) to meet per-subscriber QoS. Existing traffic-monitoring solutions [4] require additional equipment that captures every packet at every interface of a S-GW, and provides a summary to a backend SQL server every few minutes. These measurements provide no real-time visibility into the eNodeB, S-GW, and MME.

notify

The packet-handling rules in SDN switches include byte and packet counters. By adjusting these rules over time, the cellular provider can efficiently monitor traffic at different levels of granularity to drive real-time control loops on the SDN controller. In addition, associating packet classifiers with traffic counters is useful to drive billing decisions and determine whether a subscriber has reached a usage cap. The recent interest in allowing content providers to cover usage charges for mobile users will put even more pressure on cellular providers to collect fine-grained measurements.

C. Seamless Subscriber Mobility

Cellular networks must respond quickly to subscriber mobility to avoid disruptions in service. Yet, today's cellular providers do not have direct control over routing, or common protocols for controlling forwarding across different cellular technologies (e.g., 3G, LTE, WiMax, and WiFi). As a result, handoff across technologies involves complex procedures that lead to longer delays and higher packet loss rates.

comm protocols

SDN would provide a common control protocol (e.g., OpenFlow) that works across different cellular technologies, making mobility management much easier. In addition, rather than performing hop-by-hop signaling to create a new session, the controller can push new forwarding rules to multiple switches at the same time for a lower set-up delay.

D. QoS and Access Control Policies

In today's networks, the packet gateway is the central point for fine-grained policy enforcement and charging based on the subscriber profile, application, and usage. The P-GW classifies packets based on the 5-tuple of the TCP/IP header and either drops packets (if they violate a firewall policy) or map them into QoS classes. These QoS classes further map into the Diff-Serve Code Points (DSCP) when the packets traverses IP networks en route to the base station. At the base station, only simple policies such as a maximum rate are enforced. This leads to scalability problems at the P-GW, and missed opportunities to optimize the use of bandwidth inside the cellular network.

SDN would enable the distributed enforcement of QoS and firewall policies based on a network-wide view. Distributed enforcement is especially important for handling any traffic that stays within the cellular network. A controller application running on the controller can spread access-control rules over multiple switches, and manage the scheduling of traffic by QoS classes across multiple hops in the network.

E. Virtual Cellular Operators

Today's cellular networks have relatively limited support for virtualization. LTE can isolate different enterprise customers' traffic into virtual private networks using traditional BGP/MPLS VPN technologies. However, LTE does not allow different carriers to share the infrastructure to offer a complete virtual LTE network to their customers. Virtual operators may want to innovate in mobility management, policy, and charging, without investing the substantial resources necessary to build and manage a wireless network. For example, content providers like Akamai could leverage a virtual infrastructure to better deliver content to mobile users.

Virtualization would also be useful to provide isolation and separate control for different classes of traffic. For example, a carrier may want to carry traffic for roaming subscribers on a different virtual network from its own customers, for security reasons.

SDN makes it relatively easy to support network virtualization by partitioning the "flow space" of packet headers. Different controller applications can manage rules acting on each portion of flow space, enabling customized control while ensuring isolation [5]. Virtualizing the cellular network requires virtualizing the base stations [6], [7], by slicing resources at the physical layer (physical channels), link layer (scheduling), or network layer (traffic shaping).

F. Inter-Cell Interference Management

In LTE, every base station can use all subcarriers. However, base stations need to coordinate their subcarrier allocations to avoid harmful interference among users. Currently, inter-cell interference management is done in a distributed fashion. A base station tells its neighboring base stations if it needs to use a higher power to transmit to a UE using a subset of subcarriers. Alternatively, upon perceiving high interference on certain subcarriers, the base station can notify its neighboring base stations to lower their transmission power. Inter-cell interference management algorithm is related to graph coloring algorithm. Due to the lack of a global view, distributed graph coloring algorithms with a limited number of rounds are highly suboptimal [8].

SDN enables centralized control of base stations. A SDN controller will have a global view of the current power and subcarrier allocation profile of base stations. In addition, a SDN controller running on a commodity server would have much more computing resources than most base stations. As a result, a SDN controller can make a more efficient allocation of radio resources to handle new users.

III. Cellular SDN Architecture

Cellular networks need an SDN architecture that offers fine-grain, real-time control without sacrificing scalability. Working our way down from the controller platforms to the base stations, we propose four main extensions to enable SDN in cellular networks. The cellular SDN architecture is shown in Figure 3. First, controller applications should be able to express policy in terms of subscriber attributes, rather than IP addresses or physical locations, as captured in a subscriber

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0057

information base. Second, to improve control-plane scalability, each switch should run a local control agent that performs simple actions (such as polling traffic counters and comparing against a threshold), at the behest of the controller. Third, switches should support more flexible data-plane functionality, such as deep packet inspection and header compression. Fourth, base stations should support remote control of virtualized wireless resources to enable flexible cell management.

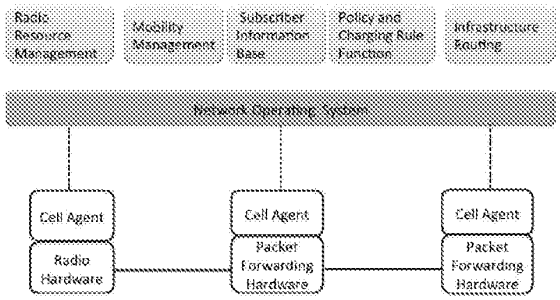


Fig. 3. Cellular SDN architecture

A. Controller: Policies on Subscriber Attributes

The SDN controller consists of a Network Operating System (NOS) running a collection of application modules, such as radio resource management, mobility management, and routing. The handling of individual packets often depends on multiple modules. For example, the flow of traffic through the network depends on the subscriber's location (determined by the mobility manager) and the paths between pairs of network elements (determined by infrastructure routing), and traffic monitoring and packet scheduling depend on the policy and charging rule function. As such, the NOS should support *composition* to combine the results of multiple modules into a single set of packet-handling rules in each switch [9].

Many of the controller application modules need to apply policy based on the properties of cellular subscribers, including the cellular network provider (e.g., whether the user is roaming or not), device type (e.g., whether the user has a legacy phone that requires echo cancellation), subscriber type (e.g., usage cap, parental controls, etc.), and recent usage (e.g., whether the user is near exceeding the usage cap). Yet, the switches match packets and perform actions based on packet header fields, based on ephemeral identifiers such as a subscriber's current IP address and location. To bridge the gap, the controller can maintain a Subscriber Information Base (SIB) that stores and maintains subscriber information, including relatively static subscriber attributes as well as dynamic data like the user's current IP address, location, and total traffic consumption.

The NOS can translate policies expressed in terms of subscriber attributes into switch rules that match on packet headers. Similarly, the NOS can translate network measurements (such as traffic counters) to the appropriate (sets of) subscribers, to allow the application modules to focus on subscribers and their attributes rather than ephemeral network identifiers[1]. The controller can also dynamically divide the network into "slices" that handle all traffic matching some predicate on the subscriber attributes. This allows the cellular provider to isolate roaming traffic, isolate dumb phone traffic or phone traffic using legacy protocols. To enable scalable slicing of semantic space, the controller can instruct ingress switches to mark incoming packets (e.g., using an MPLS label or VLAN tag) sent to or from subscribers with particular attributes.

B. Switch Software: Local Control Agents

Cellular data networks face significant scalability challenges, in terms of the number of subscribers, frequent changes in user location, fine-grain access-control and quality-of-service policies, and real-time adaptation to network conditions. For example, the switches may need to direct a video stream through a transcoding proxy if the network becomes congested, or give certain traffic lower priority if the user exceeds his usage cap. These measurement and control functions could easily overwhelm a logically-centralized controller. In addition, the controller may not be able to respond as quickly to local events as the underlying switches themselves. This argues for having some control-plane functionality on the underlying switches, though arguably not the same complex software that runs on the switches in today's cellular networks.

In particular, each switch can run an agent that performs simple local actions, under the command of the controller. For example, the controller could offload simple measurement tasks to the local agents, such as periodically polling the traffic counters and notifying the controller if a counter exceeds a threshold. The local agent could also perform simple control operations, such as automatically changing the weight or priority of a queue when traffic counts exceed a threshold, or pushing a tag on a packet to direct the traffic through an intermediate middlebox. Performing these operations on the local agents would reduce the load on the controller, and enable faster responses to critical events.

event & notify

Supporting local agents on the switches raises many interesting research problems. Partitioning functionality between the controller and the agents requires ways for application modules to expose the inherent parallelism across agents and the necessary aggregation of information at the controller. In addition, the partitioning of functionality must work in the presence of multiple modules that form a single application. Network updates must be kept consistent amid the partitioning of functionality and user equipment mobility. We plan to explore the design of the local agent and techniques for partitioning functionality in our future work.

C. Switch Hardware: Flexible Packet Processing

Today's OpenFlow [11] switches already support many features needed in cellular networks. Flexible packet classification based on Ethernet, IP, and TCP and UDP header fields enables

[1]This is similar to the way Ethane [10] supports access-control policies based on named principals, rather than IP addresses and network locations, though cellular networks must support a much wider range of attributes and control actions.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0058

fine-grain quality-of-service, access control, and monitoring. The forwarding actions in today's OpenFlow switches would enable carriers to direct selective traffic through middleboxes, change the paths to and from a mobile user, and mark and schedule traffic according to QoS policies. The byte and packet counters associated with each rule would support traffic measurement, real-time adaptation based on congestion or exceeding a subscriber's usage cap, and usage-based billing. Still, these switches may need larger rule tables, or more stages of tables, than today's commodity switches to efficiently support fine-grained policies.

That said, software-defined cellular networks would benefit from new switch capabilities. TCP/UDP port numbers are no longer a sufficiently reliable way to identify applications. Instead, support for deep packet inspection (DPI) would enable finer-grain classification based on the application, such as Web, peer-to-peer, video, and VoIP traffic. This is important to divide traffic into separate traffic classes for different packet-scheduling and routing policies, as commonly done in today's cellular networks [12]. DPI would also help support intrusion detection and prevention systems that analyze packet contents to identify malicious traffic.

To support DPI, the OpenFlow protocol must be extended to add and remove rules where the pattern is a regular expression. To avoid having the DPI module inspect every packet, SDN controller can jointly manage the flow table and the DPI rule table. For example, the switch could first match a packet with a flow-table rule, which includes a flag indicating whether the packet should proceed directly to the output port(s) or go through the DPI table. To contain cost, we do not envision that *every* switch would support DPI. Instead, the controller must place packet-classification rules in the appropriate locations in the underlying network.

In addition to DPI, switches could also support techniques like header compression and decompression to reduce the overhead for applications with small packet payloads. For example, VoIP packets are typically small, making the headers a relatively high fraction of the traffic. Compressing these packets before transmission on low-bandwidth links substantially lowers the overhead. VoIP packet sizes range from 20 to 150 bytes, and the combined overhead of the RTP, UDP, and IP headers is 40 bytes. Robust header compression (ROHC) reduces the 40-byte overhead to 1 byte. As with DPI, this functionality may not be available on every switch, but instead mainly on switches with links to and from low-bandwidth regions of the network.

D. Base Station: Remote Control and Virtualization

Remote control: In LTE, base stations participate in distributed control protocols to manage radio resource allocation, session setup/reconfiguration/teardown, handoff, and paging. Radio resources are inherently shared among base stations. The lack of central control makes it difficult to optimize tasks related to radio access. We decouple the control plane from the radio hardware. We envision that that the SDN radio hardware exposes a well-defined API which can be controlled by the control plane. Running Radio Resource Management Module (RRM) on top of a logically-centralized controller makes it much easier to innovate in admission control, radio resource allocation, and interference management. For example, the RRM can redirect a UE to a nearby lightly loaded base station or increase the transmission power of a congested base station. If the base station has multiple antennas, the control plane can decide whether the antennas should be used for boosting signals (combining diversity to boost signals for delay-sensitive applications) or for spatial multiplexing (multiple parallel transmissions). Although 3G base stations are controlled by a central entity, the radio network controller (RNC) couples control-plane and data-plane functionality, including fine-grained tasks like packet scheduling. In contrast, the RRM module should only perform control-plane functions, and instruct the base station to perform any data-plane operations.

Virtualization: Today's SDN virtualization solutions like FlowVisor [5] can share a single switch data plane among multiple virtual networks. Much as a hypervisor resides between software and hardware on a PC, FlowVisor uses OpenFlow as a hardware abstraction layer to sit logically between control and forwarding paths on a network device. FlowVisor is transparent to both the network hardware and the controller managing the virtual networks. FlowVisor defines a slice as a set of flows running on a topology of switches. The virtualization layer enforces strong isolation between slices. Resources that can be sliced are bandwidth, topology, traffic, device CPU, and forwarding tables.

Technologies like FlowVisor can be extended to slice base station resources, e.g., to create virtual base stations. A virtual base station's resource can be a combination of time slots, subcarriers, and power. For example, a virtual base station can be allocated a subset of time slots, and transmission in each time slot can use maximum transmission power allowed or a subset of subcarriers across all time slots and transmission in each time slot uses a fraction of the maximum transmission power. A slice can ask for base stations with a specific MAC protocol. To support base station virtualization, without modifying the physical-layer protocols, the controller can convey high-level information like the identity of the virtual provider through the control plane. This would allow the UE software to display the virtual provider (rather than the physically broadcasted provider information) without requiring explicit broadcasting of the provider information and cell ID.

IV. Related Work

Our work is heavily inspired and follows the high-level vision of OpenRoads (OpenFlow wireless) [7], which is a platform for innovation and realistic deployment of services for wireless networks. OpenRoads is the first software-defined wireless network. It is mainly based on WiFi and offers no special support for cellular networks. In contrast, our work addresses specific cellular network requirements such as real-time session management which runs on top of SCTP instead of TCP, paging, UE (User Equipment) state tracking, policy enforcement, charging and radio resource management.

Our work focuses on the design of a programmable control plane for cellular networks. It can benefit from a pro-

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0059

grammable data plane [13]. A programmable data plane enables physical layer customization of virtual base stations that traditional MAC scheduling based virtualization techniques [6] can not. For example, each LTE virtual base station can pick a different FFT size to balance multipath effects against peak-to-average-power ratio (PAPR) inefficiencies.

V. Conclusion and Future Work

In this paper, we argue that software defined networking can make cellular networks much simpler and easier to manage, introduce new services, and inter-operate with other wireless network technologies and other operator networks. We sketch out changes and extensions to controller platforms, network switches, and base stations to enable software defined cellular networks. The complete design of an SDN architecture for cellular networks is an exciting avenue for future work.

References

[1] S. Sesia, M. Baker, and I. Toufik, *LTE - the UMTS Long Term Evolution: From Theory to Practice*. John Wiley & Sons, 2011.

[2] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream control transmission protocol," Oct 2000. RFC 2960.

[3] "Skyfire launches its Rocket 2.0 platform to help mobile operators deal with a growing data tsunami." http://skyfire.com/images/press_releases/skyfirerocketoptimizerpress.pdf.

[4] "Alcatel-Lucent 9900 wireless network guardian." http://www.alcatel-lucent.com/wps/portal/products/detail?LMSG_CABINET=Solution_Product_Catalog&LMSG_CONTENT_FILE=Products/Product_Detail_000590.xml#tabAnchor1.

[5] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Can the production network be the testbed?," in *OSDI*, Oct 2010.

[6] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A virtualization substrate for WiMAX networks," in *MOBICOM*, pp. 233–244, ACM, 2010.

[7] K.-K. Yap, R. Sherwood, M. Kobayashi, T.-Y. Huang, M. Chan, N. Handigol, N. McKeown, and G. Parulkar, "Blueprint for introducing innovation into wireless mobile networks," in *Workshop on Virtualized Infrastructure Systems and Architectures*, pp. 25–32, 2010.

[8] F. Kuhn and R. Wattenhofer, "On the complexity of distributed graph coloring," in *PODC*, pp. 7–15, 2006.

[9] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker, "Frenetic: A network programming language," in *ICFP*, Sep 2011.

[10] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. Gude, N. McKeown, and S. Shenker, "Rethinking enterprise network control," *Trans. on Networking*, vol. 17, Aug 2009.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM CCR*, vol. 38, no. 2, pp. 69–74, 2008.

[12] "Mobile application assurance on the Alcatel-Lucent 7750 service router mobile gateway: Optimize network resources, enrich and personalize user experiences, and monetize the services," 2011. Application Note.

[13] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "OpenRadio: A programmable wireless dataplane," in *Hot Topics in Software Defined Networks*, pp. 109–114, 2012.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0060

OpenFlow: Enabling Innovation in Campus Networks

March 14, 2008

Nick McKeown
Stanford University

Tom Anderson
University of Washington

Hari Balakrishnan
MIT

Guru Parulkar
Stanford University

Larry Peterson
Princeton University

Jennifer Rexford
Princeton University

Scott Shenker
University of California, Berkeley

Jonathan Turner
Washington University in St. Louis

ABSTRACT

This whitepaper proposes OpenFlow: a way for researchers to run experimental protocols in the networks they use every day. OpenFlow is based on an Ethernet switch, with an internal flow-table, and a standardized interface to add and remove flow entries. Our goal is to encourage networking vendors to add OpenFlow to their switch products for deployment in college campus backbones and wiring closets. We believe that OpenFlow is a pragmatic compromise: on one hand, it allows researchers to run experiments on heterogeneous switches in a uniform way at line-rate and with high port-density; while on the other hand, vendors do not need to expose the internal workings of their switches. In addition to allowing researchers to evaluate their ideas in real-world traffic settings, OpenFlow could serve as a useful campus component in proposed large-scale testbeds like GENI. Two buildings at Stanford University will soon run OpenFlow networks, using commercial Ethernet switches and routers. We will work to encourage deployment at other schools; and We encourage you to consider deploying OpenFlow in your university network too.

Categories and Subject Descriptors

C.2 [**Internetworking**]: Routers

General Terms

Experimentation, Design

Keywords

Ethernet switch, virtualization, flow-based

1. THE NEED FOR PROGRAMMABLE NETWORKS

Networks have become part of the critical infrastructure of our businesses, homes and schools. This success has been both a blessing and a curse for networking researchers; their work is more relevant, but their chance of making an impact is more remote. The reduction in real-world impact of any given network innovation is because the enormous installed base of equipment and protocols, and the reluctance to experiment with production traffic, which have created an exceedingly high barrier to entry for new ideas. Today, there is almost no practical way to experiment with new network protocols (e.g., new routing protocols, or alternatives to IP) in sufficiently realistic settings (e.g., at scale carrying real traffic) to gain the confidence needed for their widespread deployment. The result is that most new ideas from the networking research community go untried and untested; hence the commonly held belief that the network infrastructure has "ossified".

Having recognized the problem, the networking community is hard at work developing programmable networks, such as GENI [1] a proposed nationwide research facility for experimenting with new network architectures and distributed systems. These programmable networks call for programmable switches and routers that (using *virtualization*) can process packets for multiple isolated experimental networks simultaneously. For example, in GENI it is envisaged that a researcher will be allocated a *slice* of resources across the whole network, consisting of a portion of network links, packet processing elements (e.g. routers) and end-hosts; researchers program their slices to behave as they wish. A slice could extend across the backbone, into access networks, into college campuses, industrial research labs, and include wiring closets, wireless networks, and sensor networks.

Virtualized programmable networks could lower the barrier to entry for new ideas, increasing the rate of innovation in the network infrastructure. But the plans for nationwide facilities are ambitious (and costly), and it will take years for them to be deployed.

This whitepaper focuses on a shorter-term question closer to home: *As researchers, how can we run experiments in our campus networks?* If we can figure out how, we can start soon and extend the technique to other campuses to benefit the whole community.

To meet this challenge, several questions need answering, including: In the early days, how will college network administrators get comfortable putting experimental equipment (switches, routers, access points, etc.) into their network? How will researchers control a portion of their local network in a way that does not disrupt others who depend on it? And exactly what functionality is needed in network switches to enable experiments? Our goal here is to propose a new switch feature that can help extend programmability into the wiring closet of college campuses.

One approach - that we do not take - is to persuade

commercial "name-brand" equipment vendors to provide an open, programmable, virtualized platform on their switches and routers so that researchers can deploy new protocols, while network administrators can take comfort that the equipment is well supported. This outcome is very unlikely in the short-term. Commercial switches and routers do not typically provide an open software platform, let alone provide a means to virtualize either their hardware or software. The practice of commercial networking is that the standardized external interfaces are narrow (i.e., just packet forwarding), and all of the switch's internal flexibility is hidden. The internals differ from vendor to vendor, with no standard platform for researchers to experiment with new ideas. Further, network equipment vendors are understandably nervous about opening up interfaces inside their boxes: they have spent years deploying and tuning fragile distributed protocols and algorithms, and they fear that new experiments will bring networks crashing down. And, of course, open platforms lower the barrier-to-entry for new competitors.

A few open software platforms already exist, but do not have the performance or port-density we need. The simplest example is a PC with several network interfaces and an operating system. All well-known operating systems support routing of packets between interfaces, and open-source implementations of routing protocols exist (e.g., as part of the Linux distribution, or from XORP [2]); and in most cases it is possible to modify the operating system to process packets in almost any manner (e.g., using Click [3]). The problem, of course, is performance: A PC can neither support the number of ports needed for a college wiring closet (a fanout of 100+ ports is needed per box), nor the packet-processing performance (wiring closet switches process over 100Gbits/s of data, whereas a typical PC struggles to exceed 1Gbit/s; and the gap between the two is widening).

Existing platforms with specialized hardware for line-rate processing are not quite suitable for college wiring closets either. For example, an ATCA-based virtualized programmable router called the Supercharged PlanetLab Platform [4] is under development at Washington University, and can use network processors to process packets from many interfaces simultaneously at line-rate. This approach is promising in the long-term, but for the time being is targeted at large switching centers and is too expensive for widespread deployment in college wiring closets. At the other extreme is NetFPGA [5] targeted for use in teaching and research labs. NetFPGA is a low-cost PCI card with a user-programmable FPGA for processing packets, and 4-ports of Gigabit Ethernet. NetFPGA is limited to just four network interfaces—insufficient for use in a wiring closet.

Thus, the commercial solutions are too closed and inflexible, and the research solutions either have insufficient performance or fanout, or are too expensive. It seems unlikely that the research solutions, with their complete generality, can overcome their performance or cost limitations. A more promising approach is to compromise on generality and to seek a degree of switch flexibility that is:

- Amenable to high-performance and low-cost implementations.
- Capable of supporting a broad range of research.
- Assured to isolate experimental traffic from production traffic.

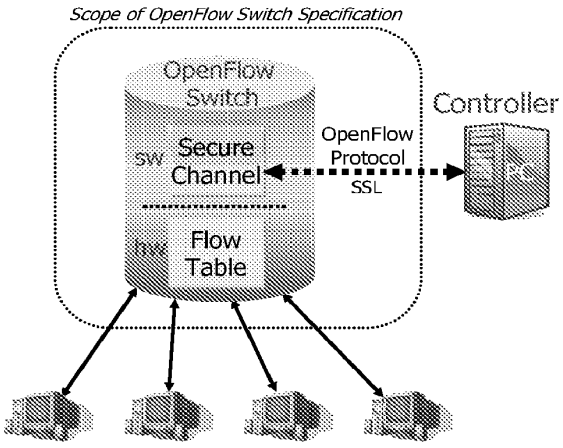


Figure 1: Idealized OpenFlow Switch. The Flow Table is controlled by a remote controller via the Secure Channel.

- Consistent with vendors' need for closed platforms.

This paper describes the OpenFlow Switch a specification that is an initial attempt to meet these four goals.

2. THE OPENFLOW SWITCH

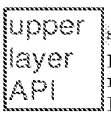
The basic idea is simple: we exploit the fact that most modern Ethernet switches and routers contain flow-tables (typically built from TCAMs) that run at line-rate to implement firewalls, NAT, QoS, and to collect statistics. While each vendor's flow-table is different, we've identified an interesting common set of functions that run in many switches and routers. OpenFlow exploits this common set of functions.

OpenFlow provides an open protocol to program the flow-table in different switches and routers. A network administrator can partition traffic into production and research flows. Researchers can control their own flows - by choosing the routes their packets follow and the processing they receive. In this way, researchers can try new routing protocols, security models, addressing schemes, and even alternatives to IP. On the same network, the production traffic is isolated and processed in the same way as today.

The datapath of an OpenFlow Switch consists of a Flow Table, and an action associated with each flow entry. The set of actions supported by an OpenFlow Switch is extensible, but below we describe a minimum requirement for all switches. For high-performance and low-cost the datapath must have a carefully prescribed degree of flexibility. This means forgoing the ability to specify arbitrary handling of each packet and seeking a more limited, but still useful, range of actions. Therefore, later in the paper, define a basic required set of actions for all OpenFlow switches.

An OpenFlow Switch consists of at least three parts: (1) A *Flow Table*, with an action associated with each flow entry, to tell the switch how to process the flow, (2) A *Secure Channel* that connects the switch to a remote control process (called the *controller*), allowing commands and packets

to be sent between a controller and the switch using (3) The *OpenFlow Protocol*, which provides an open and standard way for a controller to communicate with a switch. By specifying a standard interface (the OpenFlow Protocol) through which entries in the Flow Table can be defined externally, the OpenFlow Switch avoids the need for researchers to program the switch.



It is useful to categorize switches into dedicated OpenFlow switches that do not support normal Layer 2 and Layer 3 processing, and OpenFlow-enabled general purpose commercial Ethernet switches and routers, to which the OpenFlow Protocol and interfaces have been added as a new feature.

Dedicated OpenFlow switches. A dedicated OpenFlow Switch is a dumb datapath element that forwards packets between ports, as defined by a remote control process. Figure 1 shows an example of an OpenFlow Switch.

In this context, flows are broadly defined, and are limited only by the capabilities of the particular implementation of the Flow Table. For example, a flow could be a TCP connection, or all packets from a particular MAC address or IP address, or all packets with the same VLAN tag, or all packets from the same switch port. For experiments involving non-IPv4 packets, a flow could be defined as all packets matching a specific (but non-standard) header.

Each flow-entry has a simple action associated with it; the three basic ones (that all dedicated OpenFlow switches must support) are:

1. Forward this flow's packets to a given port (or ports). This allows packets to be routed through the network. In most switches this is expected to take place at line-rate.

2. Encapsulate and forward this flow's packets to a controller. Packet is delivered to Secure Channel, where it is encapsulated and sent to a controller. Typically used for the first packet in a new flow, so a controller can decide if the flow should be added to the Flow Table. Or in some experiments, it could be used to forward all packets to a controller for processing.

3. Drop this flow's packets. Can be used for security, to curb denial of service attacks, or to reduce spurious broadcast discovery traffic from end-hosts.

An entry in the Flow-Table has three fields: (1) A packet header that defines the flow, (2) The action, which defines how the packets should be processed, and (3) Statistics, which keep track of the number of packets and bytes for each flow, and the time since the last packet matched the flow (to help with the removal of inactive flows).

In the first generation "Type 0" switches, the flow header is a 10-tuple shown in Table 1. A TCP flow could be specified by all ten fields, whereas an IP flow might not include the transport ports in its definition. Each header field can be a wildcard to allow for aggregation of flows, such as flows in which only the VLAN ID is defined would apply to all traffic on a particular VLAN.

The detailed requirements of an OpenFlow Switch are defined by the *OpenFlow Switch Specification* [6].

OpenFlow-enabled switches. Some commercial switches, routers and access points will be enhanced with

In	VLAN	Ethernet			IP			TCP	
Port	ID	SA	DA	Type	SA	DA	Proto	Src	Dst

Table 1: The header fields matched in a "Type 0" OpenFlow switch.

the OpenFlow feature by adding the Flow Table, Secure Channel and OpenFlow Protocol (we list some examples in Section 5). Typically, the Flow Table will re-use existing hardware, such as a TCAM; the Secure Channel and Protocol will be ported to run on the switch's operating system. Figure 2 shows a network of OpenFlow-enabled commercial switches and access points. In this example, all the Flow Tables are managed by the same controller; the OpenFlow Protocol allows a switch to be controlled by two or more controllers for increased performance or robustness.

Our goal is to enable experiments to take place in an existing production network alongside regular traffic and applications. Therefore, to win the confidence of network administrators, OpenFlow-enabled switches must isolate experimental traffic (processed by the Flow Table) from production traffic that is to be processed by the normal Layer 2 and Layer 3 pipeline of the switch. There are two ways to achieve this separation. One is to add a *fourth action*:

4. Forward this flow's packets through the switch's normal processing pipeline.

The other is to define separate sets of VLANs for experimental and production traffic. Both approaches allow normal production traffic that isn't part of an experiment to be processed in the usual way by the switch. All OpenFlow-enabled switches are required to support one approach or the other; some will support both.

Additional features. If a switch supports the header formats and the four basic actions mentioned above (and detailed in the *OpenFlow Switch Specification*), then we call it a "Type 0" switch. We expect that many switches will support additional actions, for example to rewrite portions of the packet header (e.g., for NAT, or to obfuscate addresses on intermediate links), and to map packets to a priority class. Likewise, some Flow Tables will be able to match on arbitrary fields in the packet header, enabling experiments with new non-IP protocols. As a particular set of features emerges, we will define a "Type 1" switch.

Controllers. A controller adds and removes flow-entries from the Flow Table on behalf of experiments. For example, a static controller might be a simple application running on a PC to statically establish flows to interconnect a set of test computers for the duration of an experiment. In this case the flows resemble VLANs in current networks—providing a simple mechanism to isolate experimental traffic from the production network. Viewed this way, OpenFlow is a generalization of VLANs.

One can also imagine more sophisticated controllers that dynamically add/remove flows as an experiment progresses. In one usage model, a researcher might control the complete network of OpenFlow Switches and be free to decide how all flows are processed. A more sophisticated controller might support multiple researchers, each with different accounts and permissions, enabling them to run multiple independent experiments on different sets of flows. Flows identified

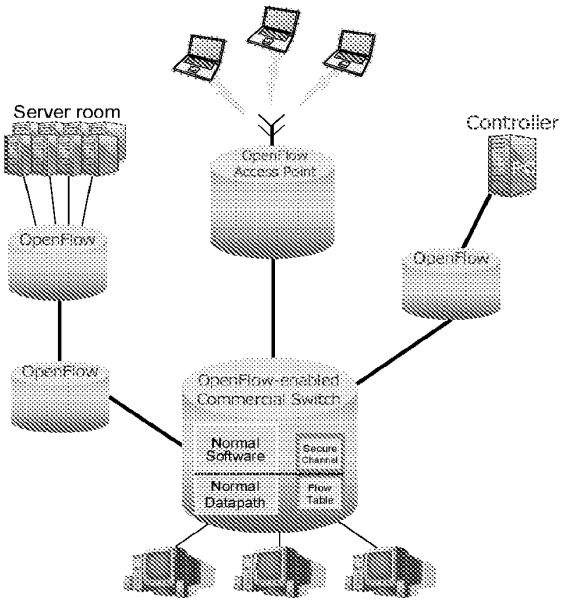


Figure 2: Example of a network of OpenFlow-enabled commercial switches and routers.

as under the control of a particular researcher (e.g., by a policy table running in a controller) could be delivered to a researcher's user-level control program which then decides if a new flow-entry should be added to the network of switches.

3. USING OPENFLOW

As a simple example of how an OpenFlow Switch might be used imagine that Amy (a researcher) invented Amy-OSPF as a new routing protocol to replace OSPF. She wants to try her protocol in a network of OpenFlow Switches, without changing any end-host software. Amy-OSPF will run in a controller; each time a new application flow starts Amy-OSPF picks a route through a series of OpenFlow Switches, and adds a flow- entry in each switch along the path. In her experiment, Amy decides to use Amy-OSPF for the traffic entering the OpenFlow network from her own desktop PC—so she doesn't disrupt the network for others. To do this, she defines one flow to be all the traffic entering the OpenFlow switch through the switch port her PC is connected to, and adds a flow-entry with the action "Encapsulate and forward all packets to a controller". When her packets reach a controller, her new protocol chooses a route and adds a new flow-entry (for the application flow) to every switch along the chosen path. When subsequent packets arrive at a switch, they are processed quickly (and at line-rate) by the Flow Table.

There are legitimate questions to ask about the performance, reliability and scalability of a controller that dynamically adds and removes flows as an experiment progresses: Can such a centralized controller be fast enough to process new flows and program the Flow Switches? What happens when a controller fails? To some extent these questions were addressed in the context of the Ethane prototype, which used simple flow switches and a central controller [7]. Preliminary results suggested that an Ethane controller based on a low-cost desktop PC could process over 10,000 new flows per second — enough for a large college campus. Of course, the rate at which new flows can be processed will depend on the complexity of the processing required by the researcher's experiment. But it gives us confidence that meaningful experiments can be run. Scalability and redundancy are possible by making a controller (and the experiments) stateless, allowing simple load-balancing over multiple separate devices.

3.1 Experiments in a Production Network

Chances are, Amy is testing her new protocol in a network used by lots of other people. We therefore want the network to have two additional properties:

1. Packets belonging to users other than Amy should be routed using a standard and tested routing protocol running in the switch or router from a "name-brand" vendor.

2. Amy should only be able to add flow entries for her traffic, or for any traffic her network administrator has allowed her to control.

Property 1 is achieved by OpenFlow-enabled switches. In Amy's experiment, the default action for all packets that don't come from Amy's PC could be to forward them through the normal processing pipeline. Amy's own packets would be forwarded directly to the outgoing port, without being processed by the normal pipeline.

Property 2 depends on the controller. The controller should be seen as a platform that enables researchers to implement various experiments, and the restrictions of Property 2 can be achieved with the appropriate use of permissions or other ways to limit the powers of individual researchers to control flow entries. The exact nature of these permission-like mechanisms will depend on how the controller is implemented. We expect that a variety of controllers will emerge. As an example of a concrete realization of a controller, some of the authors are working on a controller called NOX as a follow-on to the Ethane work [8]. A quite different controller might emerge by extending the GENI management software to OpenFlow networks.

3.2 More Examples

As with any experimental platform, the set of experiments will exceed those we can think of up-front — most experiments in OpenFlow networks are yet to be thought of. Here, for illustration, we offer some examples of how OpenFlow-enabled networks could be used to experiment with new network applications and architectures.

Example 1: Network Management and Access Control. We'll use Ethane as our first example [7] as it was the research that inspired OpenFlow. In fact, an OpenFlow Switch can be thought of as a generalization of Ethane's datapath switch. Ethane used a specific implementation of a controller, suited for network management and control, that manages the admittance and routing of flows. The basic idea of Ethane is to allow network managers to define a

network-wide policy in the central controller, which is enforced directly by making admission control decisions for each new flow. A controller checks a new flow against a set of rules, such as "Guests can communicate using HTTP, but only via a web proxy" or "VoIP phones are not allowed to communicate with laptops." A controller associates packets with their senders by managing all the bindings between names and addresses — it essentially takes over DNS, DHCP and authenticates all users when they join, keeping track of which switch port (or access point) they are connected to. One could envisage an extension to Ethane in which a policy dictates that particular flows are sent to a user's process in a controller, hence allowing researcher-specific processing to be performed in the network.

Example 2: VLANs. OpenFlow can easily provide users with their own isolated network, just as VLANs do. The simplest approach is to statically declare a set of flows which specify the ports accessible by traffic on a given VLAN ID. Traffic identified as coming from a single user (for example, originating from specific switch ports or MAC addresses) is tagged by the switches (via an action) with the appropriate VLAN ID.

A more dynamic approach might use a controller to manage authentication of users and use the knowledge of the users' locations for tagging traffic at runtime.

Example 3: Mobile wireless VOIP clients. For this example consider an experiment of a new call- handoff mechanism for WiFi-enabled phones. In the experiment VOIP clients establish a new connection over the OpenFlow-enabled network. A controller is implemented to track the location of clients, re-routing connections — by reprogramming the Flow Tables — as users move through the network, allowing seamless handoff from one access point to another.

Example 4: A non-IP network. So far, our examples have assumed an IP network, but OpenFlow doesn't require packets to be of any one format — so long as the Flow Table is able to match on the packet header. This would allow experiments using new naming, addressing and routing schemes. There are several ways an OpenFlow-enabled switch can support non-IP traffic. For example, flows could be identified using their Ethernet header (MAC src and dst addresses), a new EtherType value, or at the IP level, by a new IP Version number. More generally, we hope that future switches will allow a controller to create a generic mask (offset + value + mask), allowing packets to be processed in a researcher-specified way.

Example 5: Processing packets rather than flows. The examples above are for experiments involving flows — where a controller makes decisions when the flow starts. There are, of course, interesting experiments to be performed that require every packet to be processed. For example, an intrusion detection system that inspects every packet, an explicit congestion control mechanism, or when modifying the contents of packets, such as when converting packets from one protocol format to another.

There are two basic ways to process packets in an OpenFlow-enabled network. First, and simplest, is to force all of a flow's packets to pass through a controller. To do this, a controller doesn't add a new flow entry into the Flow Switch — it just allows the switch to default to forwarding every packet to a controller. This has the advantage of flexibility, at the cost of performance. It might provide a useful way to test the functionality of a new protocol, but is unlikely to be of much interest for deployment in a large network.

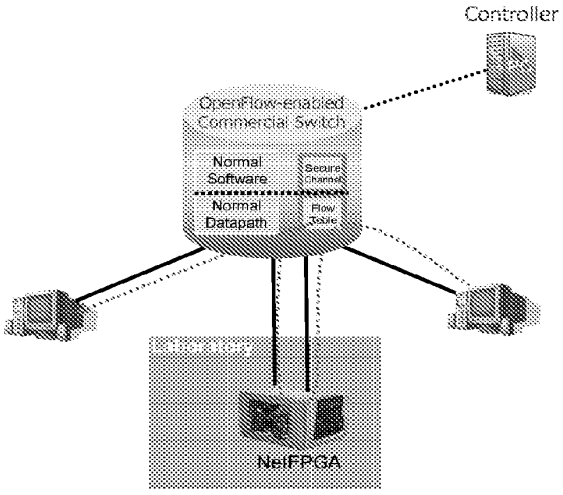


Figure 3: Example of processing packets through an external line-rate packet-processing device, such as a programmable NetFPGA router.

The second way to process packets is to route them to a programmable switch that does packet processing — for example, a NetFPGA-based programmable router. The advantage is that the packets can be processed at line-rate in a user-definable way; Figure 3 shows an example of how this could be done, in which the OpenFlow-enabled switch operates essentially as a patch-panel to allow the packets to reach the NetFPGA. In some cases, the NetFPGA board (a PCI board that plugs into a Linux PC) might be placed in the wiring closet alongside the OpenFlow-enabled switch, or (more likely) in a laboratory.

4. THE OPENFLOW CONSORTIUM

The OpenFlow Consortium aims to popularize OpenFlow and maintain the *OpenFlow Switch Specification*. The Consortium is a group of researchers and network administrators at universities and colleges who believe their research mission will be enhanced if OpenFlow-enabled switches are installed in their network.

Membership is open and free for anyone at a school, college, university, or government agency worldwide. The OpenFlow Consortium welcomes individual members who are not employed by companies that manufacture or sell Ethernet switches, routers or wireless access points (because we want to keep the consortium free of vendor influence). To join, send email to *join@OpenFlowSwitch.org*.

The Consortium web-site [1] contains the OpenFlow Switch Specification, a list of consortium members, and reference implementations of OpenFlow switches.

[1] http://www.OpenFlowSwitch.org

Licensing Model: The OpenFlow Switch Specification is free for all commercial and non-commercial use. (The exact wording is on the web-site.) Commercial switches and routers claiming to be "OpenFlow-enabled" must conform to the requirements of an OpenFlow Type 0 Switch, as defined in the OpenFlow Switch Specification. OpenFlow is a trademark of Stanford University, and will be protected on behalf of the Consortium.

5. DEPLOYING OPENFLOW SWITCHES

We believe there is an interesting market opportunity for network equipment vendors to sell OpenFlow-enabled switches to the research community. Every building in thousands of colleges and universities contains wiring closets with Ethernet switches and routers, and with wireless access points spread across campus.

We are actively working with several switch and router manufacturers who are adding the OpenFlow feature to their products by implementing a Flow Table in existing hardware; i.e. no hardware change is needed. The switches run the Secure Channel software on their existing processor.

We have found network equipment vendors to be very open to the idea of adding the OpenFlow feature. Most vendors would like to support the research community without having to expose the internal workings of their products.

We are deploying large OpenFlow networks in the Computer Science and Electrical Engineering departments at Stanford University. The networks in two buildings will be replaced by switches running OpenFlow. Eventually, all traffic will run over the OpenFlow network, with production traffic and experimental traffic being isolated on different VLANs under the control of network administrators. Researchers will control their own traffic, and be able to add/remove flow-entries.

We also expect many different OpenFlow Switches to be developed by the research community. The OpenFlow website contains "Type 0" reference designs for several different platforms: Linux (software), OpenWRT (software, for access points), and NetFPGA (hardware, 4-ports of 1GE). As more reference designs are created by the community we will post them. We encourage developers to test their switches against the reference designs.

All reference implementations of OpenFlow switches posted on the web site will be open-source and free for commercial and non-commercial use.[2]

[2]Some platforms may limit the license terms of software running on them. For example, a reference implementation on Linux may be limited by the Linux GPL.

6. CONCLUSION

We believe that OpenFlow is a pragmatic compromise that allows researchers to run experiments on heterogeneous switches and routers in a uniform way, without the need for vendors to expose the internal workings of their products, or researchers to write vendor-specific control software.

If we are successful in deploying OpenFlow networks in our campusses, we hope that OpenFlow will gradually catch-on in other universities, increasing the number of networks that support experiments. We hope that a new generation of control software emerges, allowing researchers to re-use controllers and experiments, and build on the work of others. And over time, we hope that the islands of OpenFlow networks at different universities will be interconnected by tunnels and overlay networks, and perhaps by new OpenFlow networks running in the backbone networks that connect universities to each other.

7. REFERENCES

[1] Global Environment for Network Innovations. Web site http://geni.net.

[2] Mark Handley Orion Hodson Eddie Kohler. "XORP: An Open Platform for Network Research," *ACM SIGCOMM Hot Topics in Networking, 2002.*

[3] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. "The Click modular router," *ACM Transactions on Computer Systems 18(3), August 2000, pages 263-297.*

[4] J. Turner, P. Crowley, J. Dehart, A. Freestone, B. Heller, F. Kuhms, S. Kumar, J. Lockwood, J. Lu, M.Wilson, C. Wiseman, D. Zar. "Supercharging PlanetLab - High Performance, Multi-Application, Overlay Network Platform," *ACM SIGCOMM '07, August 2007, Kyoto, Japan.*

[5] NetFPGA: Programmable Networking Hardware. Web site http://netfpga.org.

[6] The OpenFlow Switch Specification. Available at http://OpenFlowSwitch.org.

[7] Martin Casado, Michael J. Freedman, Justin Pettit, Jianying Luo, Nick McKeown, Scott Shenker. "Ethane: Taking Control of the Enterprise," *ACM SIGCOMM '07, August 2007, Kyoto, Japan.*

[8] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martin Casadao, Nick McKeown, Scott Shenker, "NOX: Towards an Operating System for Networks," *In submission. Also: http://nicira.com/docs/nox-nodis.pdf.*

Network Functions Virtualisation

An Introduction, Benefits, Enablers, Challenges & Call for Action

OBJECTIVES

This is a non-proprietary white paper authored by network operators.

The key objective for this white paper is to outline the benefits, enablers and challenges for Network Functions Virtualisation (as distinct from Cloud/SDN) and the rationale for encouraging an international collaboration to accelerate development and deployment of interoperable solutions based on high volume industry standard servers.

CONTRIBUTING ORGANISATIONS & AUTHORS

AT&T:	Margaret Chiosi.
BT:	Don Clarke, Peter Willis, Andy Reid.
CenturyLink:	James Feger, Michael Bugenhagen, Waqar Khan, Michael Fargano.
China Mobile:	Dr. Chunfeng Cui, Dr. Hui Deng.
Colt:	Javier Benitez.
Deutsche Telekom:	Uwe Michel, Herbert Damker.
KDDI:	Kenichi Ogaki, Tetsuro Matsuzaki.
NTT:	Masaki Fukui, Katsuhiro Shimano.
Orange:	Dominique Delisle, Quentin Loudier, Christos Kolias.
Telecom Italia:	Ivano Guardini, Elena Demaria, Roberto Minerva, Antonio Manzalini.
Telefonica:	Diego López, Francisco Javier Ramón Salguero.
Telstra:	Frank Ruhl.
Verizon:	Prodip Sen.

PUBLICATION DATE

October 22-24, 2012 at the "SDN and OpenFlow World Congress", Darmstadt-Germany.

This white paper is available at the following link: http://portal.etsi.org/NFV/NFV_White_Paper.pdf

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0067

Contents

Executive Summary 3
Introduction 5
Definition 5
Relationship with Software Defined Networks (SDN) 5
Fields of Application and Use Cases 6
Benefits of Network Functions Virtualisation 8
The Changing Telecoms Industry Landscape 9
Enablers for Network Functions Virtualisation 9
Cloud Computing 9
Industry Standard High Volume Servers 10
Challenges for Network Functions Virtualisation 10
Recommendations/Call for Action 12
Contact Information 13
References 14
Glossary 14

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0068

Executive Summary

Network Operators' networks are populated with a large and increasing variety of proprietary hardware appliances. To launch a new network service often requires yet another variety and finding the space and power to accommodate these boxes is becoming increasingly difficult; compounded by the increasing costs of energy, capital investment challenges and the rarity of skills necessary to design, integrate and operate increasingly complex hardware-based appliances. Moreover, hardware-based appliances rapidly reach end of life, requiring much of the procure-design-integrate-deploy cycle to be repeated with little or no revenue benefit. Worse, hardware lifecycles are becoming shorter as technology and services innovation accelerates, inhibiting the roll out of new revenue earning network services and constraining innovation in an increasingly network-centric connected world.

Network Functions Virtualisation aims to address these problems by leveraging standard IT virtualisation technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage, which could be located in Datacentres, Network Nodes and in the end user premises. We believe Network Functions Virtualisation is applicable to any data plane packet processing and control plane function in fixed and mobile network infrastructures.

We would like to emphasise that we see Network Functions Virtualisation as highly complementary to Software Defined Networking (SDN). These topics are mutually beneficial but are not dependent on each other. Network Functions can be virtualised and deployed without an SDN being required and vice-versa.

Virtualising Network Functions could potentially offer many benefits including, but not limited to:

- Reduced equipment costs and reduced power consumption through consolidating equipment and exploiting the economies of scale of the IT industry.
- Increased speed of Time to Market by minimising the typical network operator cycle of innovation. Economies of scale required to cover investments in hardware-based functionalities are no longer applicable for software-based development, making feasible other modes of feature evolution. Network Functions Virtualisation should enable network operators to significantly reduce the maturation cycle.
- Availability of network appliance multi-version and multi-tenancy, which allows use of a single platform for different applications, users and tenants. This allows network operators to share resources across services and across different customer bases.
- Targeted service introduction based on geography or customer sets is possible. Services can be rapidly scaled up/down as required.
- Enables a wide variety of eco-systems and encourages openness. It opens the virtual appliance market to pure software entrants, small players and academia, encouraging more innovation to bring new services and new revenue streams quickly at much lower risk.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0069

To leverage these benefits, there are a number of technical challenges which need to be addressed:

- Achieving high performance virtualised network appliances which are portable between different hardware vendors, and with different hypervisors.
- Achieving co-existence with bespoke hardware based network platforms whilst enabling an efficient migration path to fully virtualised network platforms which re-use network operator OSS/BSS. OSS/BSS development needs to move to a model in-line with Network Functions Virtualisation and this is where SDN can play a role.
- Managing and orchestrating many virtual network appliances (particularly alongside legacy management systems) while ensuring security from attack and misconfiguration.
- Network Functions Virtualisation will only scale if all of the functions can be automated.
- Ensuring the appropriate level of resilience to hardware and software failures.
- Integrating multiple virtual appliances from different vendors. Network operators need to be able to "mix & match" hardware from different vendors, hypervisors from different vendors and virtual appliances from different vendors without incurring significant integration costs and avoiding lock-in.

The authors of this white paper believe (and have in some cases demonstrated) that solutions to these technical challenges are available, or could be made available, and recommend that the IT and Networks industries combine their complementary expertise and resources in a joint collaborative effort to reach broad agreement on standardised approaches and common architectures which address these technical challenges, and which are interoperable and have economies of scale.

To accelerate progress, a new network operator-led Industry Specification Group (ISG) with open membership is being setup under the auspices of ETSI to work through the technical challenges for Network Functions Virtualisation as outlined in this white paper. The formal creation process of this ETSI ISG has been started and is expected to be completed by mid-November 2012.

In order to chart the way forward for Network Functions Virtualisation the wider industry is asked to provide feedback as detailed in this white paper.

End - Executive Summary.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0070

Introduction

Network operators' networks are populated with a large and increasing variety of proprietary hardware appliances. To launch a new network service often requires yet another variety and finding the space and power to accommodate these boxes is becoming increasingly difficult; compounded by the increasing costs of energy, capital investment challenges and the rarity of skills necessary to design, integrate and operate increasingly complex hardware-based appliances. Moreover, hardware-based appliances rapidly reach end of life, requiring much of the procure-design-integrate-deploy cycle to be repeated with little or no revenue benefit. Worse, hardware lifecycles are becoming shorter as technology and services innovation accelerates, inhibiting the roll out of new revenue earning network services and constraining innovation in an increasingly network-centric connected world.

Definition

Network Functions Virtualisation aims to transform the way that network operators architect networks by evolving standard IT virtualisation technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage, which could be located in Datacentres, Network Nodes and in the end user premises, as illustrated in Figure 1. It involves the implementation of network functions in software that can run on a range of industry standard server hardware, and that can be moved to, or instantiated in, various locations in the network as required, without the need for installation of new equipment.

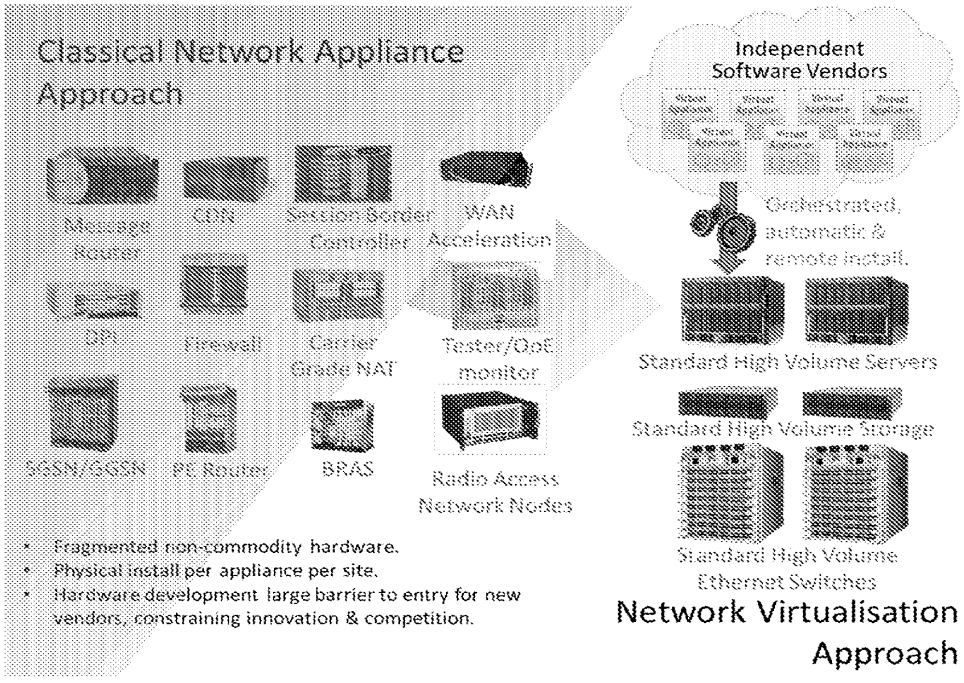


Figure 1: Vision for Network Functions Virtualisation

Relationship with Software Defined Networks (SDN)

As shown in Figure 2, Network Functions Virtualisation is highly complementary to Software Defined Networking (SDN), but not dependent on it (or vice-versa). Network Functions Virtualisation can be

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0071

implemented without a SDN being required, although the two concepts and solutions can be combined and potentially greater value accrued.

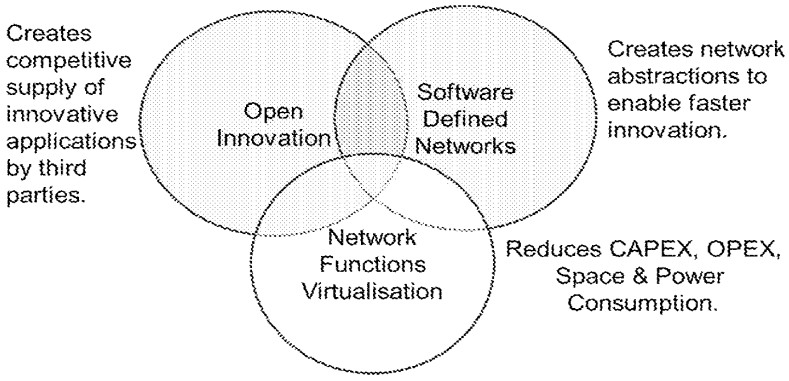


Figure 2: Network Functions Virtualisation Relationship with SDN

Network Functions Virtualisation goals can be achieved using non-SDN mechanisms, relying on the techniques currently in use in many datacentres. But approaches relying on the separation of the control and data forwarding planes as proposed by SDN can enhance performance, simplify compatibility with existing deployments, and facilitate operation and maintenance procedures.

Network Functions Virtualisation is able to support SDN by providing the infrastructure upon which the SDN software can be run. Furthermore, Network Functions Virtualisation aligns closely with the SDN objectives to use commodity servers and switches.

We intend to work closely with organisations progressing work on SDN such as the Open Networking Foundation (ONF) whose work we will specifically take into account.

Fields of Application and Use Cases

Network Functions Virtualisation is applicable to any data plane packet processing and control plane function in mobile and fixed networks. Potential examples that can be listed include (not in any particular order):

- Switching elements: BNG, CG-NAT, routers.
- Mobile network nodes: HLR/HSS, MME, SGSN, GGSN/PDN-GW, RNC, Node B, eNode B.
- Functions contained in home routers and set top boxes to create virtualised home environments.
- Tunnelling gateway elements: IPSec/SSL VPN gateways.
- Traffic analysis: DPI, QoE measurement.
- Service Assurance, SLA monitoring, Test and Diagnostics.
- NGN signalling: SBCs, IMS.
- Converged and network-wide functions: AAA servers, policy control and charging platforms.
- Application-level optimisation: CDNs, Cache Servers, Load Balancers, Application Accelerators.
- Security functions: Firewalls, virus scanners, intrusion detection systems, spam protection.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0072

Further studies are required to identify those network functions for which virtualisation yields most benefits. Use cases of interest to the authors include (not in any particular order):

- A software-based DPI, providing advanced traffic analysis and multi-dimensional reporting, and showing the possibility of making off-the-shelf hardware work at actual line rates. Software-based DPI can be pervasively deployed in the network, providing much better analysis capabilities, as well as simpler mechanisms for deployment, update, testing, and to scale it to changing workloads.
- IP node implementations, supporting - for example, but not limited to: CG-NAT and BRAS capabilities on standard high-end servers, offering the opportunity for an effective re-use of hardware as the demand for such capabilities evolves.
- The virtualisation of services and capabilities that presently require dedicated hardware appliances on customer premises (home environment to small branch office to large corporate premises), including but not restricted to: firewall, web security, IPS/IDS, WAN acceleration and optimisation, and router functions. The virtualisation of the home environment including routers, hubs and set top boxes would potentially enable a simpler and seamless migration to IPv6, reduce energy consumption and avoid successive hardware updates as broadband applications and services evolve.
- The virtualisation of Content Distribution Networks (CDN), with the initial goal of extending and scaling Content Delivery Services more easily, and also with the objective of maximizing hardware re-use in PoPs by being able to install other Service Delivery Applications (e.g. Web Acceleration) on demand. Virtualisation of CDNs will also allow the hosting of CDN services from potential business partners, like external CDN providers.
- The virtualisation of a mobile core network targeting at a more cost efficient production environment, which allows network operators to cope with the increasing traffic demand in mobile networks, and leading to better resource utilization (including energy savings), more flexible network management (no need to change hardware for nodes' upgrades), hardware consolidation, easier multi-tenancy support and faster configuration of new services. Network Functions Virtualisation in mobile networks can also be used to create core network instances optimized for specific services, e.g. for Machine-to-Machine communications (M2M).
- Co-ordinated implementation of cloud and networking for enterprises, allowing on-demand services to be offered and providing capital efficiency for enterprise customers and network operators.
- Hybrid fibre-DSL nodes are located deep in the external network in street cabinets, underground and on poles. These nodes must be very low power consumption and very low/zero maintenance to be economic. Virtualisation could be used to reduce hardware complexity at the remote node, saving energy and providing an enhanced degree of future proofing as services evolve. These remote nodes could more economically provide both fixed and wireless access if key functions were virtualised on a common platform.
- Network Functions Virtualisation can also be used to provide an efficient production environment which can commonly be used by different applications, users and tenants, thus

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0073

supporting the coexistence of several versions and variants of a network service (including test versions and beta versions).

Benefits of Network Functions Virtualisation

We believe the application of Network Functions Virtualisation brings many benefits to network operators, contributing to a dramatic change in the telecommunications industry landscape. Benefits we foresee include (not in any particular order):

- Reduced equipment costs and reduced power consumption through consolidating equipment and exploiting the economies of scale of the IT industry (~9.5M Servers shipped in 2011 compared with ~1.5M routers forecast for 2012).[1,2]
- Increased velocity of Time to Market by minimising the typical network operator cycle of innovation. Economies of scale required to cover investments in hardware-based functionalities are no longer applicable for software-based development, making feasible other modes of feature evolution. Network Functions Virtualisation should enable network operators to significantly reduce the maturation cycle.
- The possibility of running production, test and reference facilities on the same infrastructure provides much more efficient test and integration, reducing development costs and time to market.
- Targeted service introduction based on geography or customer sets is possible. Services can be rapidly scaled up/down as required. In addition, service velocity is improved by provisioning remotely in software without any site visits required to install new hardware.
- Enabling a wide variety of eco-systems and encouraging openness. It opens the virtual appliance market to pure software entrants, small players and academia, encouraging more innovation to bring new services and new revenue streams quickly at much lower risk.
- Optimizing network configuration and/or topology in near real time based on the actual traffic/mobility patterns and service demand. For example, optimisation of the location & assignment of resources to network functions automatically and in near real time could provide protection against failures without engineering full 1+1 resiliency.
- Supporting multi-tenancy thereby allowing network operators to provide tailored services and connectivity for multiple users, applications or internal systems or other network operators, all co-existing on the same hardware with appropriate secure separation of administrative domains.
- Reduced energy consumption by exploiting power management features in standard servers and storage, as well as workload consolidation and location optimisation. For example, relying on virtualisation techniques it would be possible to concentrate the workload on a smaller number of servers during off-peak hours (e.g. overnight) so that all the other servers can be switched off or put into an energy saving mode.[3]
- Improved operational efficiency by taking advantage of the higher uniformity of the physical network platform and its homogeneity to other support platforms:

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0074

- IT orchestration mechanisms provide automated installation, scaling-up and scaling-out of capacity, and re-use of Virtual Machine (VM) builds.[4]
- Eliminating the need for application-specific hardware. The skills base across the industry for operating standard high volume IT servers is much larger and less fragmented than for today's telecom-specific network equipment.
- Reduction in variety of equipment for planning & provisioning. Assuming tools are developed for automation and to deal with the increased software complexity of virtualisation.
- Option to temporarily repair failures by automated re-configuration and moving network workloads onto spare capacity using IT orchestration mechanisms. This could be used to reduce the cost of 24/7 operations by mitigating failures automatically.
- The potential to gain more efficiency between IT and Network Operations.
- The potential to support in-service software upgrade (ISSU) with easy reversion by installing the new version of a Virtualised Network Appliance (VNA) as a new Virtual Machine (VM). Assuming traffic can be transferred from the old VM to the new VM without interrupting service. For some applications it may be necessary to synchronise the state of the new VM with the old VM.

The Changing Telecoms Industry Landscape

Although Network Functions Virtualisation brings many advantages to the telecommunications industry it is likely to transform the vendor landscape. Each player will need to position/re-position itself in the new Network Functions Virtualisation market.

This is not as disruptive as it may seem because network equipment vendors already implement some of their solutions by combining their proprietary software with industry standard hardware and software components, but in a proprietary way. Enabling their proprietary software to run on industry standard hardware in a standardised way may be a significant opportunity for existing players because their software and networking know-how is where the real value is in many cases. Some major industry players are already moving in this direction by offering virtualised versions of their products. The challenge for network operators is how to migrate their operations and skill base to a software based networking environment while carefully re-targeting investment to maximise re-use of existing systems and processes.

Enablers for Network Functions Virtualisation

Several recent technology developments make the goals of Network Functions Virtualisation achievable. This section describes these enablers and briefly discusses relevance.

Cloud Computing

Network Functions Virtualisation will leverage modern technologies such as those developed for cloud computing. At the core of these cloud technologies are virtualisation mechanisms: hardware virtualisation by means of hypervisors, as well as the usage of virtual Ethernet switches (e.g. vswitch) for connecting traffic between virtual machines and physical interfaces. For communication-oriented

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0075

functions, high-performance packet processing is available through high-speed multi-core CPUs with high I/O bandwidth, the use of smart Ethernet NICs for load sharing and TCP Offloading, and routing packets directly to Virtual Machine memory, and poll-mode Ethernet drivers (rather than interrupt driven, for example Linux NAPI and Intel's DPDK).

Cloud infrastructures provide methods to enhance resource availability and usage by means of orchestration and management mechanisms, applicable to the automatic instantiation of virtual appliances in the network, to the management of resources by assigning virtual appliances to the correct CPU core, memory and interfaces, to the re-initialisation of failed VMs, to snapshot VM states and the migration of VMs.

Finally, the availability of open APIs for management and data plane control, like OpenFlow, OpenStack, OpenNaaS or OGF's NSI, provide an additional degree of integration of Network Functions Virtualisation and cloud infrastructure.

Industry Standard High Volume Servers

The use of industry standard high volume servers is a key element in the economic case for Network Functions Virtualisation. Network Functions Virtualisation leverages the economies of scale of the IT industry. An industry standard high volume server is a server built using standardised IT components (for example x86 architecture) and sold in the millions. A common feature of industry standard high volume servers is that there is competitive supply of the subcomponents which are interchangeable inside the server.

We believe that Network Appliances which depend on the development of bespoke Application Specific Integrated Circuits (ASICs) will become increasingly uncompetitive against general purpose processors as the cost of developing ASICs increases exponentially with decreasing feature size.[5] Merchant silicon will still be applicable for commodity functions implemented at scale, and ASICs will still be applicable for some types of very high throughput applications.

Challenges for Network Functions Virtualisation

There are a number of challenges to implement Network Functions Virtualisation which need to be addressed by the community interested in accelerating progress. How this effort could be progressed is described later in this document. Challenges we have identified are (not in any particular order):-

- **Portability/Interoperability.** The ability to load and execute virtual appliances in different but standardised datacentre environments, provided by different vendors for different operators. The challenge is to define a unified interface which clearly decouples the software instances from the underlying hardware, as represented by virtual machines and their hypervisors. Portability and Interoperability is very important as it creates different ecosystems for virtual appliance vendors and datacentre vendors, while both ecosystems are clearly coupled and depend on each other. Portability also allows the operator the freedom to optimise the location and required resources of the virtual appliances without constraints.
- **Performance Trade-Off.** Since the Network Functions Virtualisation approach is based on industry standard hardware (i.e. avoiding any proprietary hardware such as acceleration

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0076

engines) a probable decrease in performance has to be taken into account. The challenge is how to keep the performance degradation as small as possible by using appropriate hypervisors and modern software technologies, so that the effects on latency, throughput and processing overhead are minimized. The available performance of the underlying platform needs to be clearly indicated, so that virtual appliances know what they can get from the hardware. The authors of the white paper believe that using the right technology choice will allow virtualisation not only of network control functions but also data/user plane functions.

- **Migration and co-existence of legacy & compatibility with existing platforms.** Implementations of Network Functions Virtualisation must co-exist with network operators' legacy network equipment and be compatible with their existing Element Management Systems, Network Management Systems, OSS and BSS, and potentially existing IT orchestration systems if Network Functions Virtualisation orchestration and IT orchestration are to converge. The Network Functions Virtualisation architecture must support a migration path from today's proprietary physical network appliance based solutions to more open standards based virtual network appliance solutions. In other words, Network Functions Virtualisation must work in a hybrid network composed of classical physical network appliances and virtual network appliances. Virtual appliances must therefore use existing North Bound Interfaces (for management & control) and interwork with physical appliances implementing the same functions.
- **Management and Orchestration.** A consistent management and orchestration architecture is required. Network Functions Virtualisation presents an opportunity, through the flexibility afforded by software network appliances operating in an open and standardised infrastructure, to rapidly align management and orchestration North Bound Interfaces to well defined standards and abstract specifications. This will greatly reduce the cost and time to integrate new virtual appliances into a network operator's operating environment. Software Defined Networking (SDN) further extends this to streamlining the integration of packet and optical switches into the system e.g. a virtual appliance or Network Functions Virtualisation orchestration system may control the forwarding behaviours of physical switches using SDN.
- **Automation.** Network Functions Virtualisation will only scale if all of the functions can be automated. Automation of process is paramount to success.
- **Security & Resilience.** Network operators need to be assured that the security, resilience and availability of their networks are not impaired when virtualised network functions are introduced. Our initial expectations are that Network Functions Virtualisation improves network resilience and availability by allowing network functions to be recreated on demand after a failure. A virtual appliance should be as secure as a physical appliance if the infrastructure, especially the hypervisor and its configuration, is secure. Network operators will be seeking tools to control and verify hypervisor configurations. They will also require security certified hypervisors and virtual appliances.
- **Network Stability.** Ensuring stability of the network is not impacted when managing and orchestrating a large number of virtual appliances between different hardware vendors and hypervisors. This is particularly important when, for example, virtual functions are relocated, or during re-configuration events (e.g. due to hardware and software failures) or due to cyber-attack. This challenge is not unique to Network Functions Virtualisation. Potential instability might also occur in current networks, depending on unwanted combinations of diverse control and optimization mechanisms, for example acting on either the underlying transport network or on the higher layers' components (e.g. flow admission control, congestion control, dynamic routing and allocations, etc.). It should be noted that

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0077

occurrence of network instability may have primary effects, such as jeopardizing, even dramatically, performance parameters or compromising an optimized use of resources. Mechanisms capable of ensuring network stability will add further benefits to Network Functions Virtualisation.

- **Simplicity.** Ensuring that virtualised network platforms will be simpler to operate than those that exist today. A significant and topical focus for network operators is simplification of the plethora of complex network platforms and support systems which have evolved over decades of network technology evolution, while maintaining continuity to support important revenue generating services. It is important to avoid trading one set of operational headaches for a different but equally intractable set of operational headaches.
- **Integration.** Seamless integration of multiple virtual appliances onto existing industry standard high volume servers and hypervisors is a key challenge for Network Functions Virtualisation. Network operators need to be able to "mix & match" servers from different vendors, hypervisors from different vendors and virtual appliances from different vendors without incurring significant integration costs and avoiding lock-in. The ecosystem must offer integration services and maintenance and third-party support; it must be possible to resolve integration issues between several parties. The ecosystem will require mechanisms to validate new Network Functions Virtualisation products. Tools must be identified and/or created to address these issues.

Recommendations/Call for Action

Network Functions Virtualisation is already occurring. In a few years, we can expect the communications industry to look and feel similar to the IT industry. There will be a wider range of business models more suited to a software industry. Operations complexity will be abstracted away by more automation and self-provisioning will be more common. As detailed in this white paper, Network Functions Virtualisation will deliver many benefits for network operators and their partners and customers whilst offering the opportunity to create new types of eco-systems (alongside traditional supply models based on preferred strategic partners) which will encourage and support rapid innovation with reduced cost and reduced risk. To reap these benefits the technical challenges, as described above, must be addressed by the industry.

The authors of this white paper believe that solutions to these technical challenges are available (or could be made available) and recommend that the IT and Network industries combine their complementary expertise and resources in a joint collaborative effort to reach broad agreement on standardised approaches and common architectures which address these technical challenges, and which are interoperable and have economies of scale.

To accelerate progress, a new network operator-led Industry Specification Group (ISG) with open membership is being setup under the auspices of ETSI to work through the technical challenges for Network Functions Virtualisation as outlined in this white paper. The formal creation process of this ETSI ISG has been started and is expected to be completed by mid-November 2012.

While ETSI will provide the organisation for this initiative, the ISG will downstream its work to all relevant organisations and seek to encourage convergence of IT and Network Standardisation efforts in this space.

The key objectives, activities and outputs of the ISG are expected to be:

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0078

- The ISG will initiate joint studies which address the technical challenges for Network Functions Virtualisation as documented in this white paper. With consensus outputs detailed in white papers openly published to the wider industry. It is expected that study items would be identified in a "Network Operator Requirements" white paper prepared by the ISG as an initial deliverable.
- Although parented under ETSI, the ISG will not be a Standards Development Organisation (SDO) in itself; it will reference the work of existing SDOs where applicable. Where there are gaps, it will encourage member contributions to SDOs requesting them to take into account the recommendations of the ISG (i.e. requesting updates to existing standards, or to create new standards if appropriate).
- The ISG will identify other expert bodies that could assist with this work, and request that they undertake studies related to their scope to help move the industry forward.
- The ISG will facilitate members to share technical learning in a collaborative way which does not compromise confidentiality.
- The first meeting of the ISG will take place in January 2013. Details will be announced as soon as possible after publication of this white paper.

In order to chart the way forward for Network Functions Virtualisation, the wider industry is asked to provide feedback:

- **Network Operators:** To state their interest in joining the ISG, thus subscribing to the targets and approach as outlined in this white paper.
- **Existing organizations and forums:** To state if they see themselves being impacted by the work addressed in the ISG, and to provide information on how they would like to interact with it.
- **Vendors:** To state their interest in joining the ISG. Vendors are furthermore requested to state the study items in which they are most interested.

Please provide your input to the originators of this white paper as soon as possible and no later than 16th November 2012. Contact details below.

Contact Information

Please provide your feedback to any of the following. Please be clear on what you are asking us to do with the information provided. **Your input must not be confidential.**

AT&T:	Margaret Chiosi, mchiosi@att.com
BT:	Don Clarke, don.clarke@bt.com Peter Willis, peter.j.willis@bt.com
CenturyLink:	James Feger, james.feger@centurylink.com Michael Bugenhagen, michael.k.bugenhagen@centurylink.com Waqar Khan, waqar.khan@centurylink.com Michael Fargano, michael.fargano@centurylink.com

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0079

China Mobile:	Clark Chen, chenkuilin@chinamobile.com Jinri Huang, huangjinri@chinamobile.com
Colt:	Javier Benitez, javier.benitez@colt.net
Deutsche Telekom:	Uwe Michel, uwe.michel@telekom.de Herbert Damker, herbert.damker@telekom.de
KDDI:	Kenichi Ogaki, ke-oogaki@kddi.com
NTT:	Masaki Fukui, fukui.masaki@lab.ntt.co.jp Katsuhiro Shimano, shimano.katsuhiro@lab.ntt.co.jp
Orange:	Dominique Delisle, dominique.delisle@orange.com Quentin Loudier, quentin.loudier@orange.com Christos Kolias, christos.kolias@orange.com
Telecom Italia:	Ivano Guardini, ivano.guardini@telecomitalia.it Elena Demaria, elena.demaria@telecomitalia.it
Telefonica:	Diego López, diego@tid.es Francisco Javier Ramón Salguero, fjrs@tid.es
Telstra:	Frank Ruhl, frank.ruhl@team.telstra.com
Verizon:	Prodip Sen, prodip.sen@verizon.com

References

1. Server shipments (9.5M units in 2011): http://www.crn.com/news/data-center/232601815/gartner-2011-server-sales-grow-but-hard-drive-issues-take-a-toll.htm
2. Router shipments (1.5M units forecast for year 2012): http://www.isuppli.com/Manufacturing-and-Pricing/MarketWatch/Pages/EMS-Router-Shipments-Backed-by-Market-Growth-in-2012.aspx
3. http://labs.chinamobile.com/cran/wp-content/uploads/CRAN_white_paper_v2_5_EN.pdf
4. http://en.wikipedia.org/wiki/Orchestration_(computing)
5. http://www.edac.org/downloads/resources/profitability/HandelJonesReport.pdf

Glossary

AAA	Authentication-Authorisation-Accounting
API	Application Programming Interface (a set of routines, protocols, and tools for building software applications)
ASIC	Application Specific Integrated Circuit
BNG	Broadband Network Gateway
BRAS	Broadband Remote Access Server
BSS	Business Support System
CDN	Content Distribution Network
CG-NAT/NAT	Carrier Grade Network Address Translation
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DPDK	Data Plane Development Kit

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0080

DPI	Deep Packet Inspection
DSL	Digital Subscriber Line (typically ADSL2/+, VDSL2, G.fast)
Ecosystem	Equipment and software vendors, integration services vendors, network operators, service providers, academia, etc.
EMS	Element Management System
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute (in practise ETSI has global membership, including the Americas and Asia) .
HSS	Home Subscriber Service
HV	Hypervisor
I/O	Input/Output
IMS	IP Multimedia System
Industry Standard High Volume Server	A server built using standardised IT components (for example x86 architecture) and sold in the millions.
IPR	Intellectual Property Rights
IPSec/SSL	IP Security/ Secure Sockets Layer
IPv6	Internet Protocol version 6
ISG	Industry Specification Group. An organisation under the auspices of the European Telecommunications Standards Institute (ETSI).
ISSU	In Service Software Upgrade
IT	Information Technology
KVM	Kernel based Virtual Machine
LAN	Local Area Network
LB	Load Balancer
M2M	Machine-to-Machine communications
MME	Mobility Management Entity
MVNO	Mobile Virtual Network Operator
NFV	Network Functions Virtualisation
NGN	Next Generation Network
NIC	Network Interface Controller
NMS	Network Management System
OAM	Operations Administration & Maintenance
OGF	Open Grid Forum
ONF	Open Networking Foundation
OpenFlow	Specifications developed by the Open Networking Foundation
OpenNaaS	Specifications developed by the OpenNaaS community
OpenStack	Specifications developed by the OpenStack Foundation
OSS	Operations Support System
PDN-GW	Packet Data Network Gateway
PE Router	Provider Edge Router
PoP	Point of Presence
QoE	Quality of Experience
SBC	Session Border Controller
SDN	Software Defined Network
SDO	Standards Development Organisation
SGSN/GGSN	Serving GPRS support node/Gateway GPRS support node
SLA	Service Level Agreement

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0081

TCP	Transmission Control Protocol
VM	Virtual Machine
VMWare	Proprietary Hypervisor
VNA	Virtualised Network Appliance
VPN	Virtual Private Network
vswitch	Any Ethernet switch implemented in software alongside or inside a hypervisor. There are proprietary and open implementations of vswitch.
WAN	Wide Area Network
Xen	Proprietary Hypervisor

END

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0082



A SHORT INTRODUCTION TO CLOUD PLATFORMS

AN ENTERPRISE-ORIENTED VIEW

DAVID CHAPPELL

AUGUST 2008

SPONSORED BY MICROSOFT CORPORATION

COPYRIGHT © 2008 CHAPPELL & ASSOCIATES

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0083

CONTENTS

Defining Terms: What is a Cloud Platform? 3
Cloud Platforms in Context: Three Kinds of Cloud Services 3
A General Model for Application Platforms 4
From On-Premises Platforms to Cloud Platforms 7
Examining Cloud Platforms 7
Cloud Foundation 8
Operating System 8
Local Support 8
Cloud Infrastructure Services 9
Storage 9
Integration 10
Identity 11
Cloud Application Services 11
SaaS Application Services 11
Search 12
Mapping 12
Other Application Services 12
Conclusion 13
About the Author 13

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0084

DEFINING TERMS: WHAT IS A CLOUD PLATFORM?

The coming shift to cloud computing is a major change in our industry. One of the most important parts of that shift is the advent of *cloud platforms*. As its name suggests, this kind of platform lets developers write applications that run in the cloud, or use services provided from the cloud, or both. Different names are used for this kind of platform today, including *on-demand platform* and *platform as a service (PaaS)*. Whatever it's called, this new way of supporting applications has great potential.

To see why, think about how application platforms are used today. When a development team creates an on-premises application (i.e., one that will run within an organization), much of what that application needs already exists. An operating system provides basic support for executing the application, interacting with storage, and more, while other computers in the environment offer services such as remote storage. If the creators of every on-premises application first had to build all of these basics, we'd have many fewer applications today.

Similarly, if every development team that wishes to create a cloud application must first build its own cloud platform, we won't see many cloud applications. Fortunately, vendors are rising to this challenge, and a number of cloud platform technologies are available today. The goal of this overview is to categorize and briefly describe those technologies as they're seen by someone who creates enterprise applications.

CLOUD PLATFORMS IN CONTEXT: THREE KINDS OF CLOUD SERVICES

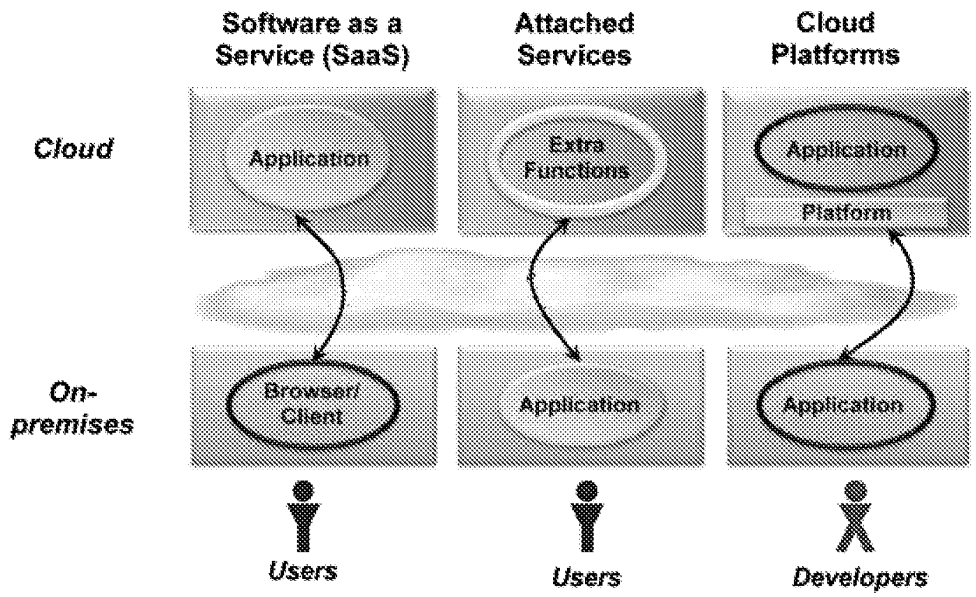


Figure 1: Cloud services can be grouped into three broad categories.

To get a grip on cloud platforms, it's useful to start by looking at cloud services in general. As Figure 1 shows, services in the cloud can be grouped into three broad categories. Those categories are:

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0085

Software as a service (SaaS): A SaaS application runs entirely in the cloud (that is, on servers at an Internet-accessible service provider). The on-premises client is typically a browser or some other simple client. The most well-known example of a SaaS application today is probably Salesforce.com, but many, many others are also available.

Attached services: Every on-premises application provides useful functions on its own. An application can sometimes enhance these by accessing application-specific services provided in the cloud. Because these services are usable only by this particular application, they can be thought of as attached to it. One popular consumer example of this is Apple's iTunes: The desktop application is useful for playing music and more, while an attached service allows buying new audio and video content. Microsoft's Exchange Hosted Services provides an enterprise example, adding cloud-based spam filtering, archiving, and other services to an on-premises Exchange server.

Cloud platforms: A cloud platform provides cloud-based services for creating applications. Rather than building their own custom foundation, for example, the creators of a new SaaS application could instead build on a cloud platform. As Figure 1 shows, the direct users of a cloud platform are developers, not end users.

Understanding cloud platforms requires some agreement on what the word "platform" means in this context. One broad way to think about it is to view a platform as any software that provides developer-accessible services for creating applications. The next section looks at this idea in a bit more detail.

A GENERAL MODEL FOR APPLICATION PLATFORMS

Our experience with application platforms today comes mostly from on-premises platforms. A useful way to think about cloud platforms is to see how the services an application developer relies on in the on-premises environment translate to the cloud. To help do this, Figure 2 shows a general model that can be applied to both worlds.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0086

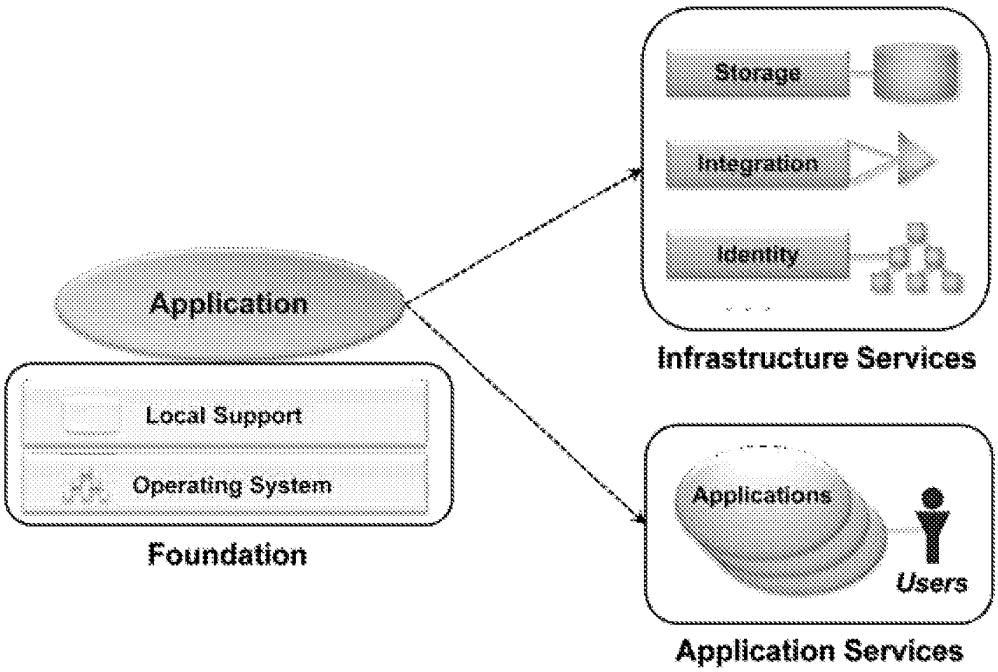


Figure 2: A modern application platform can be viewed as having three parts.

Whether it's on-premises or in the cloud, an application platform can be thought of as comprising three parts:

A *foundation*: Nearly every application uses some platform software on the machine it runs on. This typically includes various support functions, such as standard libraries and storage, and a base operating system.

A group of *infrastructure services*: In a modern distributed environment, applications frequently use basic services provided on other computers. It's common to provide remote storage, for example, integration services, an identity service, and more.

A set of *application services*: As more and more applications become service-oriented, the functions they offer become accessible to new applications. Even though these applications exist primarily to provide services to end users, this also makes them part of the application platform. (It might seem odd to think of other applications as part of the platform, but in a service-oriented world, they certainly are.)

And while they're not shown in Figure 2, development tools are another important part of this story. Modern tools can help developers build applications using all three parts of an application platform.

To make this abstract model more concrete, think about how it fits with today's most popular on-premises platforms. The on-premises foundation looks like this:

Operating system: The dominant choices are Windows, Linux, and other versions of Unix.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0087

Local support: Different technologies are used depending on the style of application. The .NET Framework and Java EE application servers provide general support for Web applications and more, for instance, while other technologies target specific kinds of applications. For example, Microsoft's Dynamics CRM product includes a platform designed for creating a particular type of business application. Similarly, different kinds of storage are used for different purposes. Raw byte storage is provided by the file systems in Windows, Linux, and other operating systems, while more structured storage is provided by a range of database technologies, including the Oracle DBMS, MySQL, Microsoft SQL Server, and IBM DB2.

For on-premises infrastructure services, typical examples include the following:

Storage: Like storage in the foundation, infrastructure storage comes in various styles. A remote file system might provide simple byte-oriented storage, while a Microsoft SharePoint document library provides a more structured remote storage service. Applications can also access a database system remotely, allowing access to another kind of structured storage.

Integration: Connecting applications within an organization usually depends on a remote service provided by some integration product. A message queue is a simple example of this, while more complex scenarios use products such as IBM's WebSphere Process Server, Microsoft's BizTalk Server, and others.

Identity: Providing identity information is a fundamental requirement for most distributed applications. Common on-premises technologies that address this include Microsoft's Active Directory and other LDAP servers.

On-premises application services, the third category shown in Figure 2, vary widely across different organizations. The reason for this is simple: Different organizations use different applications, which in turn expose diverse services. One way to think about these applications in the on-premises platform is to divide them into two broad categories:

Packaged applications: This includes business software such as SAP, Oracle Applications, and Microsoft Dynamics, along with a myriad of other off-the-shelf products. While not all packaged applications expose services to other applications, more and more of them do.

Custom applications: Many organizations have a large investment in custom software. As these applications increasingly expose their functionality through services, they become part of the on-premises application platform.

When it's described like this, the on-premises application platform can seem quite complex. The truth, though, is that this platform has evolved over time. In the early days of computing, the application platform consisted of nothing more than an on-premises foundation. (Think of MVS and IMS on an IBM mainframe, for example.) In the 1980s and 1990s, as distributed computing spread, on-premises infrastructure services were added, with remote storage, integration, and identity becoming common. Today, with the advent of service-oriented applications, on-premises application services have become part of the platform. The next step in this evolution is clear: providing cloud versions of all three.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0088

FROM ON-PREMISES PLATFORMS TO CLOUD PLATFORMS

Along with describing on-premises platforms, the general model just described can also be used to think about cloud platforms. And since on-premises and cloud platforms can be used together, it's important to understand how the two work in concert. Figure 3 illustrates this new world.

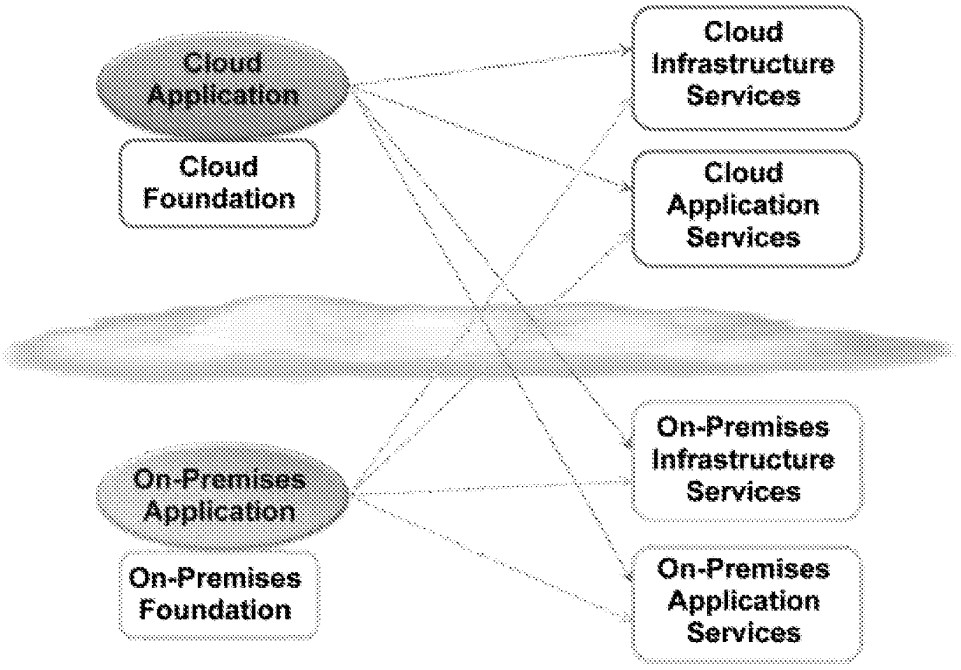


Figure 3: On-premises platforms and cloud platforms can be viewed in similar ways, and they can also be used together.

As the figure shows, a cloud application can be built on a cloud foundation, just as an on-premises application is built on an on-premises foundation. Both kinds of applications can access infrastructure and application services provided on-premises and in the cloud. Just as on-premises platforms support today's applications, cloud platforms provide services for the applications we're likely to build tomorrow.

EXAMINING CLOUD PLATFORMS

Understanding cloud platforms means looking at each of their parts: the cloud foundation, cloud infrastructure services, and cloud application services. This section walks through these three areas, using some of today's most visible cloud platform technologies as examples.

Before we begin, one important note: While it's useful to look at on-premises platforms and cloud platforms through the same lens, the two aren't identical. When platform functions move into the cloud, they sometimes change in significant ways. For example, on-premises platforms are designed to support (at most) enterprise-scale applications. Applications that run in the cloud, by contrast, can potentially operate at Internet scale, which requires handling many more simultaneous users than any enterprise application. While the same kinds of platform functions might be needed in both cases, achieving this high

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0089

scalability can force a cloud platform to provide them in a quite different way. In what follows, expect to see some differences from the on-premises world.

CLOUD FOUNDATION

Like their on-premises cousins, cloud foundations provide the basic local functions an application needs. These can include an underlying operating system and local support. Yet how cloud platforms provide these functions differs from what we're used to, as this section shows.

Operating System

From a platform point of view, an operating system provides a set of basic interfaces for applications to use. By far the most well-known example of an operating system in the cloud today is Amazon's Elastic Compute Cloud (EC2). EC2 provides customer-specific Linux instances running in virtual machines (VMs). From a technical perspective, it might be more accurate to think of EC2 as a platform for VMs rather than operating systems. Still, a developer sees an operating system interface, and so viewing it in this light makes more sense here.

Each development team is free to use whatever local support it likes in this VM—Amazon doesn't care. The creators of one application might choose a Java EE app server and MySQL, for example, while another group might go with Ruby on Rails. EC2 customers are even free to create many Linux instances, then distribute large workloads across them in parallel, such as for scientific applications. While the service EC2 provides is quite basic, it's also very general, and so it can be used in many different ways.

Local Support

In an on-premises platform (and in EC2), a developer can mix and match parts of the foundation as she sees fit. Choosing to use the .NET Framework on Windows doesn't mandate using a particular database, for example. Similarly, an on-premises application using the .NET Framework is free to access the underlying Windows operating system, as is an application built on a Java EE server.

The local support functions in today's leading cloud foundations don't work this way. Instead, a cloud local support technology typically includes its own storage, and it hides whatever the underlying operating system might be. A developer choosing to build on a particular local support option must accept the limitations it imposes.

There are good reasons for these limitations, of course. One of the things that makes cloud computing so attractive is its potential for scalability, but to make an application built on a cloud foundation handle Internet-size loads requires limiting it in some ways. By making the local support functions more specialized, a cloud platform provider has more freedom to optimize the application environment. Accordingly, each set of local support functions in cloud foundations today focuses on supporting a particular kind of application.

For example, Google's AppEngine provides local support for running Python Web applications. Along with a standard Python runtime, AppEngine also includes a hierarchical data store with its own query language. Another example of a cloud platform providing local support is Force.com, offered by Salesforce.com. Rather than targeting general Web applications, however, Force.com is aimed at creating data-oriented

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0090

business applications. Toward this end, it provides its own focused support for data storage. And rather than adopt an existing programming language, this platform's creators invented their own, a language called Apex.

Microsoft also provides local support for applications in the cloud as part of its CRM Live offering. Based on the Dynamics CRM platform mentioned earlier, this technology targets data-oriented business applications, much like Force.com. And like both Force.com and AppEngine, it includes both run-time application support and a data store. Microsoft has also talked about its plans to go further in this area, with a platform that will support standard .NET development languages and tools. The intent, Microsoft says, is to allow portability of both applications and developer skills between the company's on-premises foundation and its cloud foundation.

CLOUD INFRASTRUCTURE SERVICES

Whether they run on-premises or in the cloud, some applications don't need anything beyond a foundation. Still, many can benefit from distributed storage, common identity, and other infrastructure services. We're accustomed to having these services provided on-premises today, but analogous services are also provided in the cloud.

As Figure 3 showed, cloud infrastructure services can be accessed by applications running on either an on-premises foundation or a cloud foundation. Initially, the most common users of cloud infrastructure services will be on-premises, because there aren't yet many applications built on a cloud foundation. Over time, expect this to change, as more and more cloud-based applications also use cloud infrastructure services.

Storage

Applications commonly use some kind of local storage, which is why storage is part of both on-premises and cloud foundations. Remote storage is also useful, however, as the popularity of this service in the on-premises world shows. Accordingly, it's reasonable to expect that providing a storage service in the cloud will be attractive for many applications.

As with on-premises platforms, remote storage in the cloud comes in different styles. For example, Amazon's Simple Storage Service (S3) provides basic unstructured remote storage. The model it exposes to developers is straightforward: *objects*, which are just bunches of bytes, are stored in *buckets*. Applications can create, read, and delete objects and buckets. Objects can't be updated, however—they can only be entirely replaced. This is another example of how platform services must change to support Internet-scale usage, something that Amazon is clearly focused on. This simple but limited storage service is much easier to make scalable than a more fully featured offering would be. The trade-off is clear: Application developers get cheap storage in the cloud, but they might need to do more work in their applications to use it effectively.

Another approach to cloud storage is to support more structured data. In Microsoft's SQL Server Data Services (SSDS), for example, a *container* includes one or more *entities*, each of which holds some number of *properties*, as shown in Figure 4. An application can issue queries against a container's data with operators such as ==, !=, <, >, AND, OR, and NOT.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0091

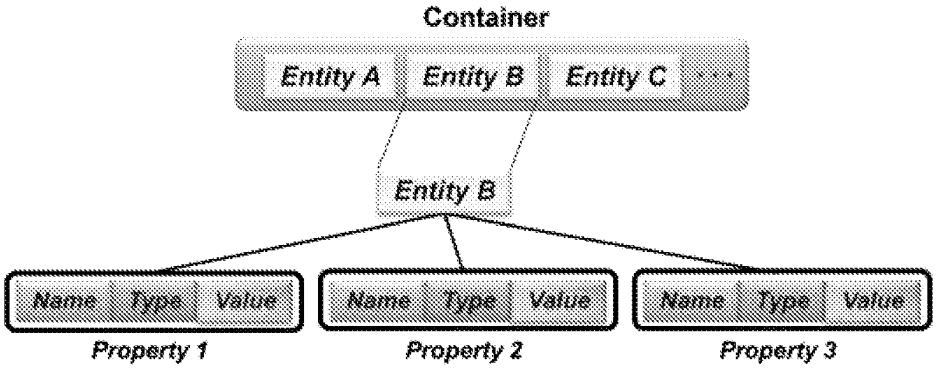


Figure 4: In SQL Server Data Services, a container holds entities with properties.

It's important to note that this isn't a relational database, and the query language isn't SQL. Once again, we're seeing an illustration of how application platform technologies change when they're moved into the cloud. This simpler approach is easier to use than a relational database—there's no need to define a schema up front—and it's also easier to make scalable.

Amazon's SimpleDB provides one more example of the value of structured storage in the cloud. The way SimpleDB organizes information is similar to SSDS—it's a hierarchy of *domains*, *items*, and *values*—and it also provides a non-SQL query language. Like SSDS, no up-front schema definition is required, and so the approach provides a mix of flexibility and scalability.

Integration

Is there any application left that doesn't talk to at least one of its fellows? Connecting applications has become a staple of computing, and vendors have provided a plethora of on-premises infrastructure services to do it. These range from relatively simple technologies like message queues to quite complex integration servers.

As integration services move into the cloud, a range of technologies is also appearing. For example, Amazon's Simple Queue Service (SQS) provides just what its name suggests: a straightforward way for applications to exchange messages via queues in the cloud. Yet SQS once again illustrates what happens when a familiar on-premises service is recast as a cloud service. Because SQS replicates messages across multiple queues, an application reading from a queue isn't guaranteed to see all messages from all queues on a particular read request. SQS also doesn't promise in-order, exactly-once delivery. These simplifications let Amazon make SQS more scalable, but they also mean that developers must use SQS differently from an on-premises message queuing technology.

BizTalk Services provides another example of cloud-based integration. Rather than using message queuing, BizTalk Services implements a relay service in the cloud that lets applications communicate through firewalls. Cloud-based integration, such as connecting applications in different organizations, typically requires traversing firewalls, and so solving this problem is important. BizTalk Services also provides simple workflow support along with a way for an application to register the services it exposes, then let those services be invoked by any other application that has permission to do so.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0092

Going forward, expect to see more integration services offered in the cloud. Given the importance of integration as an on-premises service, it shouldn't be surprising to see its functions become part of the cloud infrastructure.

Identity

Whether an application runs on-premises or in the cloud, it typically needs to know something about its users. Toward this end, the application commonly demands that each user provides a digital identity, a set of bytes that describes that user. Based on what these bytes contain and how they're verified, the application can determine things such as who this user is and what they're allowed to do.

Many on-premises applications today rely on an on-premises infrastructure service, such as Active Directory, to provide this identity information. When a user accesses a cloud application, however, or an on-premises application accesses a cloud service, an on-premises identity usually won't work. And what about an application built on a cloud foundation? Where does it get its identity information?

An identity service in the cloud can address these issues. Because it provides a digital identity that can be used by people, by on-premises applications, and by cloud applications, a cloud identity service can be applied in many different scenarios. In fact, one indication of the importance of this kind of identity service is the number of cloud identity services available today. Accessing Amazon cloud services such as EC2 or S3 requires presenting an Amazon-defined identity, for instance, while using Google AppEngine requires a Google account. Microsoft provides Windows Live ID, which can be used for Microsoft applications and others, while BizTalk Services also offers its own identity service, which can be federated with others. Developers don't have complete freedom—cloud platforms are frequently tied to a particular identity provider—but the need for identity as a cloud service is clear.

CLOUD APPLICATION SERVICES

What's the difference between an application service and an infrastructure service? To answer this question, think first about the obvious distinction between applications and infrastructure: Applications are designed to be used by people, while infrastructure is designed to be used by applications. It's also fair to say that infrastructure usually provides a general, relatively low-level service, while applications provide more specific, higher-level services. An infrastructure service solves a broad problem faced by many different kinds of applications, while an application service solves a more targeted problem. And just as it's possible to identify different kinds of infrastructure services, it's also possible to distinguish different categories of application services, as this section illustrates.

SaaS Application Services

Users in most enterprises today rely on both purchased and home-grown applications. As these applications expose their services to remote software, they become part of the on-premises platform. Similarly, SaaS applications today frequently expose services that can be accessed by on-premises applications or by other cloud applications. Salesforce.com's CRM application, for example, makes available a variety of services that can be used to integrate its functions with on-premises applications. As organizations begin to create their own SaaS applications running on a cloud foundation, those applications will also expose services. Just as packaged and custom on-premises applications today are

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0093

part of the on-premises platform, the services exposed by packaged and custom SaaS applications are becoming part of the cloud platform.

Search

Services exposed by SaaS applications are useful, but they're not the whole story. Other kinds of cloud application services are also important. Think, for example, of search engines such as Google and Live Search. Along with their obvious value to people, why can't they also offer cloud application services?

The answer, of course, is that they can. Microsoft's Live Search, for example, exposes services that allow on-premises and cloud applications to submit searches and get results back. Suppose a company that provided a database of legal information wanted to let customers search both its own data and the Web in a single request. They could accomplish this by creating an on-premises application that both searched their proprietary data and, via the Live Search application service, the entire Web. It's fair to say that not many applications are likely to need this kind of service, but that's one reason why it's most accurate to think of search as an application service rather than an infrastructure service.

Mapping

Many Web applications today display maps. Hotel Web sites plot their locations, retailers provide store locators, and more. The people who create these applications probably don't have the time, interest, or budget to create their own mapping database. Yet enough applications need this function to justify creating a cloud application service that provides it.

This is exactly what's done by mapping services such as Google Maps and Microsoft's Virtual Earth. Both provide cloud-based services that application developers can use to embed maps in Web pages and more. And as with search, these mapping services are adjuncts to existing Web sites that target users directly, i.e., they're cloud application services.

Other Application Services

Many other application services are available today. In fact, almost any Web site can expose its functionality as a cloud service for developers to use. Photo-sharing sites such as Google's Picasa and Microsoft's Windows Live Photo Gallery do this, for example, as do online contacts applications such as Google Contacts and Microsoft's Windows Live Contacts. One big motivation for exposing services is to make it easier to create mash-ups that exploit the functions of diverse Web applications.

Vendors sometimes group cloud application services together under a common umbrella. The services for accessing information in Google Contacts, Picasa, and more are all part of the Google Data APIs, for instance. Similarly, Microsoft groups several of its services together under the Live Platform brand, including Live Search, Virtual Earth, Windows Live Contacts, Windows Live ID, an Alerts service, a specialized storage service called Application-Based Storage, and several more.

The line between cloud infrastructure services and cloud application services can sometimes be fuzzy. General cloud storage services such as S3 and SSDS are clearly infrastructure, for example, as are cloud identity services. A mapping service such as Google Earth is just as clearly application-centric—only certain kinds of apps need it—as is a service like Live Search. But an Alerts service might be considered

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0094

infrastructure, since it's more generally useful, and Windows Live ID is definitely infrastructure, even though Microsoft views both services as part of its Live Platform.

Cloud platforms are a relatively new area, and so it shouldn't be surprising that defining a firm taxonomy is challenging. However you choose to view them, it's clear that cloud application services have an important role to play. Knowing what's available in the cloud should be a core competency today for everyone who designs and builds software.

CONCLUSION

A new kind of application platform doesn't come along very often. But when a successful platform innovation does appear, it has an enormous impact. Think of the way personal computers and servers shook up the world of mainframes and minicomputers, for example, or how the rise of platforms for N-tier applications changed the way people write software. While the old world doesn't go away, a new approach can quickly become the center of attention for new applications.

Cloud platforms don't yet offer the full spectrum of an on-premises environment. For example, business intelligence as part of the platform isn't common, nor is support for business process management technologies such as full-featured workflow and rules engines. This is all but certain to change, however, as this technology wave continues to roll forward.

Cloud platforms aren't yet at the center of most people's attention. The odds are good, though, that this won't be true five years from now. The attractions of cloud-based computing, including scalability and lower costs, are very real. If you work in application development, whether for a software vendor or an end user, expect the cloud to play an increasing role in your future. The next generation of application platforms is here.

ABOUT THE AUTHOR

David Chappell is Principal of Chappell & Associates (www.davidchappell.com) in San Francisco, California. Through his speaking, writing, and consulting, he helps software professionals around the world understand, use, and make better decisions about new technology.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0095

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Filer:	Christopher James Horgan
Attorney Docket Number:	TAAG001CONT3

Filed as Small Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
UTILITY FILING FEE (ELECTRONIC FILING)	4011	1	80	80
UTILITY SEARCH FEE	2111	1	350	350
UTILITY EXAMINATION FEE	2311	1	400	400
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**830**

Electronic Acknowledgement Receipt	
EFS ID:	41573796
Application Number:	17142983
International Application Number:	
Confirmation Number:	2943
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Customer Number:	112449
Filer:	Christopher James Horgan
Filer Authorized By:	
Attorney Docket Number:	TAAG001CONT3
Receipt Date:	06-JAN-2021
Filing Date:	
Time Stamp:	18:22:21
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$830
RAM confirmation Number	E202116I24496049
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	TAAG001CONT3_ADSv4.pdf	125468 c26ba63e7b8347ef9122411e4770e77b3f9a4f98	no	7
Warnings:					
Information:					
This is not an USPTO supplied ADS fillable form					
2	Power of Attorney	TAAG001CONT3_POA.pdf	213764 85c4f45ed9214d76b6c65bc872ddca03cd49b013	no	2
Warnings:					
Information:					
3	Assignee showing of ownership per 37 CFR 3.73	TAAG001CONT3_CERT.pdf	123719 2f7f54060add117eac94a2dc9694fd928e25ecf1	no	2
Warnings:					
Information:					
4	Nonpublication request from applicant	TAAG001CONT3_NONPUB.pdf	124790 42a62dc9be0f661f347edc2bcffc0246d7670592	no	1
Warnings:					
Information:					
5	Internet Communications Authorized	TAAG001CONT3_INT_COMM.pdf	69887 2d524d271ba0293765a37e702f211ee32f88701d	no	1
Warnings:					
Information:					
6	Drawings-only black and white line drawings	TAAG001CONT3_DRAWINGS_.pdf	907811 abcccd864b4c6997eb89e14f696bcb2e0a5f0bd3	no	18
Warnings:					

The page size in the PDF is too large. The pages should be 8.5 x 11 or A4. If this PDF is submitted, the pages will be resized upon entry into the Image File Wrapper and may affect subsequent processing					
Information:					
7	Specification	TAAG001CONT3_SPECIFICATION_v3.pdf	661870 167217d059a5331c194106481c0fcaa9e16391de	no	61
Warnings:					
Information:					
8	Oath or Declaration filed	TAAG001CONT3_DEC.pdf	328696 1cf9d3cd80c1b0e68ab30b96d6cfc378143b43ed	no	2
Warnings:					
Information:					
9	Information Disclosure Statement (IDS) Form (SB08)	TAAG001CONT3_IDSv3.pdf	254641 c132ccd05d78c6eb148efa342dbd206432ff8e01	no	4
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
10	Non Patent Literature	NONPATLIT__LUO.pdf	852818 c9b78214cc0e2155c2e3383fd1740cb0450e3a05	no	8
Warnings:					
Information:					
11	Non Patent Literature	NONPATLIT__STALLINGS.pdf	954039 fbfef4a62ba37b2def361f029d3a9267307dae43	no	7
Warnings:					
Information:					
12	Non Patent Literature	NONPATLIT__YAP.pdf	264023 f25648fc789a6ab17de81ad0c50fde586fd8d1a7	no	2
Warnings:					
Information:					
13	Non Patent Literature	NONPATLIT__BANSAL.pdf	692740 50119452c442a40f36be929599991991176c2498	no	6

Warnings:					
Information:					
14	Non Patent Literature	NONPATLIT__LI.pdf	924070 888b121a29ca83c98742829c788e1f2e006a44ce	no	6
Warnings:					
Information:					
15	Non Patent Literature	NONPATLIT__MCKEON.pdf	930354 8fe2f2cdcb54c1d2995b95f8dd691ac9c8c5176b	no	6
Warnings:					
Information:					
16	Non Patent Literature	NONPATLIT__NFV.pdf	883145 339ff411e0988b66d095517cf5ca34f81eef04be	no	16
Warnings:					
Information:					
17	Non Patent Literature	NONPATLIT__CHAPPELL___FINAL.pdf	779191 4017bae1920cb9702f967e200c74ab3a5b81da17	no	13
Warnings:					
Information:					
18	Fee Worksheet (SB06)	fee-info.pdf	35203 74df611717aedd66fdd44a11e31bca8ef30ff14c	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	9126229	

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.

PTO/AIA/14 (03-13)
Approved for use through 01/31/2014. 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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	TAAG001CONT3
	Application Number	
Title of Invention	Distributed Software Defined Networking	

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.)

Inventor Information:

Inventor 1 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Pouya		Taaghol	

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

City	San Jose	State/Province	CA	Country of Residence	US

Mailing Address of Inventor:

Address 1	1260 Trestlewood Lane		
Address 2			
City	San Jose	State/Province	CA
Postal Code	95138	Country	US

Inventor 2 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Vivek		Ramanna	

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

City	San Jose	State/Province	CA	Country of Residence	US

Mailing Address of Inventor:

Address 1	549 Manhattan Place		
Address 2			
City	San Jose	State/Province	CA
Postal Code	95126	Country	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).

PTO/AIA/14 (03-13)
Approved for use through 01/31/2014. 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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	TAAG001CONT3
	Application Number	
Title of Invention	Distributed Software Defined Networking	

☐ An Address is being provided for the correspondence Information of this application.			
Customer Number	112449		
Email Address	chris.horgan@roarkip.com	Add Email	Remove Email

Application Information:

Title of the Invention	Distributed Software Defined Networking		
Attorney Docket Number	TAAG001CONT3	**Small Entity Status Claimed**	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	18	**Suggested Figure for Publication (if any)**	3

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).
Either enter 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	112449		

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.

Prior Application Status	Pending		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	
	Continuation of	16900963	2020-06-14	
Prior Application Status	Pending		Remove	

PTO/AIA/14 (03-13)
Approved for use through 01/31/2014. 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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	TAAG001CONT3
	Application Number	
Title of Invention	Distributed Software Defined Networking	

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
16900963	Continuation of	15836824	2017-12-09
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
15836824	Continuation of	14295331	2014-06-04
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
14295331	non provisional of	61834807	2013-06-13

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 priority to a foreign application. 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(d). When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)[i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(h)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country[i]	Filing Date (YYYY-MM-DD)	Access Code[i] (if applicable)

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

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

PTO/AIA/14 (03-13)
Approved for use through 01/31/2014. 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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	TAAG001CONT3
	Application Number	
Title of Invention	Distributed Software Defined Networking	

Authorization to Permit Access:

☐ Authorization to Permit Access to the Instant Application by the Participating Offices

If checked, the undersigned hereby grants the USPTO authority to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the World Intellectual Property Office (WIPO), and any other intellectual property offices in which a foreign application claiming priority to the instant patent application is filed access to the instant patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, WIPO, or other intellectual property office in which a foreign application claiming priority to the instant patent application is filed to have access to the instant patent application.

In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the instant patent application with respect to: 1) the instant patent application-as-filed; 2) any foreign application to which the instant patent application claims priority under 35 U.S.C. 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the instant patent application; and 3) any U.S. application-as-filed from which benefit is sought in the instant patent application.

In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date o f filing this Authorization.

Applicant Information:

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

Applicant 1 Remove

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

◉ Assignee	○ Legal Representative under 35 U.S.C. 117	○ Joint Inventor
○ Person to whom the inventor is obligated to assign.	○ Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor :	
If the Applicant is an Organization check here. ☒	
Organization Name	Big Data Federation, Inc.

PTO/AIA/14 (03-13)
Approved for use through 01/31/2014. 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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	TAAG001CONT3
	Application Number	
Title of Invention	Distributed Software Defined Networking	

Mailing Address Information For Applicant:			
Address 1	990 Linden Drive, #220		
Address 2			
City	Santa Clara	**State/Province**	CA
Country	US	Postal Code	95050
Phone Number	408-623-8529	Fax Number	
Email Address	pouya@bigdatafeed.com		

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

Non-Applicant Assignee Information:

Providing assignment information in this section does not subsitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Assignee 1

Complete this section only if non-applicant assignee information is desired to be included on the patent application publication in accordance with 37 CFR 1.215(b). Do not include in this section an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest), as the patent application publication will include the name of the applicant(s).

Remove

If the Assignee is an Organization check here. ☐

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

Mailing Address Information For Non-Applicant Assignee:			
Address 1			
Address 2			
City		**State/Province**	
Country		Postal Code	
Phone Number		Fax Number	
Email Address			

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

PTO/AIA/14 (03-13)
Approved for use through 01/31/2014. 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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	TAAG001CONT3
	Application Number	
Title of Invention	Distributed Software Defined Networking	

Signature:

Remove

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications.					
Signature	/Christopher J. Horgan/			Date (YYYY-MM-DD)	2021-01-06
First Name	Christopher	Last Name	Horgan	Registration Number	40394
Additional Signature may be generated within this form by selecting the Add button. Add					

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.

DISTRIBUTED SOFTWARE DEFINED NETWORKING

[0001] The present invention claims priority from U.S. Patent Appln. No. 16/900,963, filed on June 14, 2020; which claims priority to U.S. Patent Appln. No. 15/836,824, filed on 12/9/2017; which claims priority from U.S. Patent Application Ser. **No.** 14/295,331, filed June 4, 2014; which claims priority from U.S. Provisional Patent Application Serial No. 61/834,807, filed June 13, 2013; both of these disclosures are herein specifically incorporated by reference in their entirety.

FIELD OF THE **INVENTION**

[0002] This invention relates to a network architecture that facilitates secure and flexible programmability between a user device and across a network with full lifecycle management of services and infrastructure applications.

BACKGROUND

[0003] Existing network solutions are built on proprietary hardware and software. Network Operators & Information Technology (NOIT) can only configure the proprietary network infrastructure provided by vendors and are unable to add new and customized features or capabilities. As a result, the NOIT can only add new desired features by making such requests to their infrastructure vendors or pursuing standardization processes. But these existing approaches are time consuming and resourceintensive.

SUMMARY OF THE INVENTION

[0004] An aspect of the disclosure herein is a plurality of network elements in a packet system, said network elements comprising: a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp; a flexible cloud platform having at least one fxCloudApp, wherein the at least one fxDeviceApp and at least one fxCloudApp are in secure communication to form a distributed application; and an application management portal which manages life cycle of the distributed applications.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0110

[0005] Another aspect of the disclosure is a plurality of network elements in a packet system, said network elements comprising: a flexible cloud platform having a plurality of virtual machines, wherein each of the virtual machines has at least one fxCloudApp; and a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp, wherein the at least one fxDeviceApp and fxCloudApp are in secure communication to form a distributed application.

[0006] Another aspect of the disclosure is a plurality of network elements in a packet system, said network elements comprising: a flexible cloud platform having a plurality of zones, wherein each of the plurality of zones has at least one fxCloudApp; and a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp, wherein the at least one fxDeviceApp and fxCloudApp are in secure communication to form a distributed application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Figure 1 demonstrates the network architecture of a 4G system.

[0008] Figure 2 illustrates a network element (or system) upon which the Distributed Software Defined Network operates.

[0009] Figure 3 illustrates a Distributed Software Defined Network in accordance with the present disclosure.

[001O] Figure 4 is an exemplary view of the software layers in the Flexible Operation System (fxOS) Architecture.

[0011] Figure 5 illustrates the FastPath architecture.

[0012] Figure 6 presents a potential software layer of the fxOS co-existing alongside a general purpose OS.

[0013] Figure 7 is an exemplary view of the fxStore Portal.

[0014] Figure 8 demonstrates a shared fxCloud deployment.

[0015] Figure 9 demonstrates a Zoned Deployment.

[0016] Figure 10 demonstrates another deployment option as applied to cellular networks (especially Long Term Evolution (LTE)).

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0111

[0017] Figure 11 shows an example of a signaling flow chart for thefxDevice Commissioning Procedure.

[0018] Figure 12 shows an example procedure for Application Provisioning.

[0019] Figure 13 presents the state machine for the application upgrade process.

[0020] Figures 14A-14C are examples of the Distributed Switch Controller during Upgrade.

[0021] Figure 15 shows how fxVF in the fxCloud and fxDevice.

[0022] Figure 16 demonstrates the logical interfaces between the platform resources and applications.

[0023] Figures 17A and 17B demonstrate example procedures for cloud breathing procedures.

DETAILED DESCRIPTION

[0024] The Distributed Software Defined Network (dSDN) disclosed herein is an architecture that enables secure and flexible programmability across a network with full lifecycle management of services and infrastructure applications (fxDeviceApp). The dSDN also harmonizes application deployment across the network independent of the hardware vendor. As a result, the dSDN simplifies the network deployment lifecycle from concept to design to implementation to decommissioning.

DEFINITIONS, ACRONYMS & ABBREVIATIONS

[0025] The following terms, acronyms, abbreviations and descriptions are explained below and are used throughout the detailed description of the dSDN:

TERM	DESCRIPTION
2G	2nd Generation Cellular Technology
3G	3rd Generation Cellular Technology
4G	4th Generation Cellular Technology
AP	WiFi Access Point
API	Application Programming Interface

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0112

AR	Application Repository
ASIC	Application Specification Integrated Circuit
AuC	Authentication Center
BS	Base Station
BSC	Base Station Controller
BTS	Base Station
CON	Content Distribution Network
CN	Core Network
CNIE	Core Network Internet Edge
CPU	Central Processing Unit
dAP	Distributed Application Package
dApp	Distributed App
dCP	Distributed Content Service
dNS	Distributed Notification Service
DNS	Domain Name System
DPDK	Data Plane Development Kit
DPI	Deep Packet Inspection
dRS	Distributed Resource Service
DSCP	Differentiated Service Code Point (QoS)
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
EIR	Equipment Identity Register
eNB	Enhanced Node B (4G)
FIB	Flow Information Base
FQDN	Fully Qualified Domain Name
fxApp	FleXible Application, which has a fxDeviceApp and fxCloudApp component. fxDeviceApp and fxCloudApp of one application represent fxApp.
fxDeviceApp	FleXible Application which runs on fxDevice
fxBS	FleXible Base Station that is a BS that is powered by fxOS (an example of an fxDevice)

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0113

fxCloud	FleXible Cloud that interacts with fxDevice and on other network elements on the northbound Interface
fxCloudApp	FleXible Cloud Application that is the application that runs on fxCloud and can be associated to one or more fxDeviceApp in fxDevice
fxManager	FleXible Manager that manager the fxOS lifecycle (provisioning, usage, de-provisioning)
fxOS	FleXible Operation System which run on the fxDevice as the OS and firmware
fxSDK:	FleXible Software Development Kit
fxSimulator	FleXible Simulator
fxStore	FleXible Store that presents fxApp to the network
administratorsGERAN	GPRS Edge Radio Access Network
GGSN	Gateway GPRS Support Node (used in core network of 2G/3G systems)uhu
GMSC	Gateway Mobile Switching Center
GPRS	General Packet Radio Service
GPS	Global Positioning System
GTP	GPRS Tunneling Protocol
GW	Gateway
HAL	Hardware Abstraction Layer
HD	High Definition
HLR	Home Location Register
HSS	Home Subscriber Server
IAC	Inter Application Communication

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0114

IMS	IP Multimedia Sub-system
IMSI	International Mobile Subscriber Identifier
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
JVM	Java Virtual Machine
LA	Location Area
LAN	Local Area Network
LSP	Label Switching Protocol
M2M	Machine to Machine
MAC	Medium Access Control
MCP	Multi Chip Packaging
MIMO	Multi Input Multi Output
MME	Mobility Management Entity (used in core network of LTE)
MNO	Mobile Operator Network
MPLS	Multi Protocol Label Switching
MSC	Mobile Switching Center
MSISDN	Mobile Station
NAS	Non-Access Stratum
NB	Node B (3G)
NFC	Near Field Communication
NFV	Network Function Virtualization
NOIT	Network Operations & Information Technology; this term network broadly includes carriers, service providers, enterprise networks, and designated/contacted 3rd party administrators.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0115

OS	Operating System
OTT	Over the Top
P-GW	Packet Gateway
PaaS	Platform-as-a-Service
PBX	Private Branch exchange (telephony)
PDG	Packet Data Gateway
PON	Packet Data Network
PHY	Physical Layer
PLMN	Public Land Mobile Network
PnP	Plug and Play
pp	Packet Processor
PR	Platform Resources
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RA	Routing Area
RAN	Radio Access Network of cellular networks
RAPI	Resources API
RFID	Radio Frequency Identifier
RNC	Radio Network Controller
S-GW	Serving Gateway
S1-AP	S1 Interface Application Protocol (an LTE protocol between MME and eNodeB)
Saas	Software-as-a-Software
SOK	Software Development Kit
SON	Software Defined Network
SGSN	Serving GPRS Support Node (used in core

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0116

	network of 2G/3G systems)
SIM	Subscriber Identity Module
SiP	System in a Package
SMS	Short Message Service
SoC	System on a Chip
SON	Self-Optimizing Network
SSDP	Simple Service Discovery Protocol
TA	Target Area
TNE	Test Network Environment
TTM	Time to Market
UMTS	Universal Mobile Telecommunication System
UPnP	Universal Plug and Play
URA	UMTS Routing Area
URI	Uniform Resource Identifier
UTRAN	UMTS Terrestrial Radio Access Network
VLR	Visitor Location Register
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WLAN	Wireless Local Area Network
XMPP	Extensible Messaging and Presence Protocol

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0117

[0026] Database. One or more large structured sets of persistent datamaintained upon a computer system organized and structured according to a software system defining rules for organization as well as responding to queries to read, write, or modify data as well as provide statistical information regarding the contained data. As used in this disclosure in describing the dSDN, a database may be either a single unified system or a distributed system wherein certain database elements are located upon different systems or servers which may be in different physical locations, acting in harmony to appear as one unified database. Where databases are described, it will be understood by one of ordinary skill in the art that (i) alternative database structures to those described may be readily employed, and (ii) other memory structures besides databases may be readily employed. Any illustrations or descriptions of any sample databases presented herein are illustrative arrangements for stored representations of information. The database formats may include relational databases, object-based models and/or distributed databases which could be used to store and manipulate the data types described herein. Likewise, object methods or behaviors of a database can be used to implement various processes, such as the described herein. In addition, the databases may, in a known manner, be stored locally or remotely from a device which accesses data in such a database.

[0027] The term "determining" and grammatical variants thereof (e.g., to determine a price, determining a value, determine an object which meets a certain criterion) is used in an extremely broad sense. The term "determining" encompasses a wide variety of actions and therefore "determining" can include calculating, computing, processing, deriving, investigating, looking up (e.g., looking up in a table, a database or another data structure), ascertaining and the like. Also, "determining" can include receiving (e.g., receiving information), accessing (e.g., accessing data in a memory) and the like. Also, "determining" can include resolving, selecting, choosing, establishing, and the like. The term "determining" does not imply certainty or absolute precision, and therefore "determining" can include estimating, extrapolating, predicting, guessing and the like. The term "determining" does not imply that mathematical processing must be performed, and does not imply that numerical methods must be used, and does not imply that an algorithm or process is used.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0118

[0028] The functionality and/or the features of a single device that is described may be alternatively embodied by one or more other devices which are described but are not explicitly described as having such functionality/features. Thus, other embodiments need not include the described device itself, but rather can include the one or more other devices which would, in those other embodiments, have such functionality/features.

[0029] Devices that are described as in "communication" with each other or "coupled" to each other need not be in continuous communication with each other or in direct physical contact, unless expressly specified otherwise. On the contrary, such devices need only transmit to each other as necessary or desirable, and may actually refrain from exchanging data most of the time. For example, a machine in communication with or coupled with another machine via the Internet may not transmit data to the other machine for long period of time (e.g. weeks at a time). In addition, devices that are in communication with or coupled with each other may communicate directly or indirectly through one or more intermediaries.

[0030] Although process (or method) steps may be described or claimed in a particular sequential order, such processes may be configured to work in different orders. Further, some steps may be performed simultaneously despite being described or implied as occurring non-simultaneously (e.g., because one step is described after the other step) unless specifically indicated. Moreover, the illustration of a process by its depiction in a drawing does not imply that the illustrated process is exclusive of other variations and modifications thereto, does not imply that the illustrated process or any of its steps are necessary to the embodiment(s), and does not imply that the illustrated process is preferred. Where a process is described in an embodiment the process may operate without any user intervention.

[0031] It will be readily apparent to one of ordinary skill in the art that the various processes of the dSDN described herein may be implemented by, e.g., appropriately programmed general purpose computer(s), special purpose computer(s) and computing device(s). Typically a processor (e.g., one or more microprocessors, one or more microcontrollers, one or more digital signal processors) will receive instructions (e.g., from a memory or like device), and execute those instructions,

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0119

thereby performing one or more processes defined by those instructions. Instructions may be embodied in, e.g., one or more computer programs, one or more scripts.

[0032] A "processor" means one or more microprocessors, central processingunits (CPUs), computing devices, controllers, microcontrollers, digital signal processors, or like devices or any combination thereof, regardless of the architecture (e.g., chip-level multiprocessing/multi-core, Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC), Microprocessor without Interlocked Pipeline Stages, pipelining configuration, or simultaneous multithreading).

[0033] Further, programs that implement methods described herein may be stored and transmitted using a variety of media (e.g., computer readable media) in a number of manners. In some embodiments, hard-wired circuitry or custom hardware may be used in place of, or in combination with, some or all of the software instructions that can implement the processes of various embodiments. Thus, various combinations of hardware and software may be used instead of software only to implement the embodiments.

[0034] The term "non-transitory computer readable medium" in this disclosure refers to any medium, a plurality of the same, or a combination of different media, that participate in providing data (e.g., instructions, data structures) which may be read by a computer, a processor or a like device. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media may include, for example, optical or magnetic disks and other persistent memory. Volatile media may include dynamic random access memory (DRAM), which typically constitutes the main memory. Transmission media may include coaxial cables, copper wire and fiber optics, including the wires that comprise a system bus coupled to the processor. Transmission media may include or convey acoustic waves, light waves and electromagnetic emissions, such as those generated during radio frequency (RF) and infrared (IR) data communications. Common forms of non-transitory computer-readable media in which the dSDN may be implemented include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, any other magnetic medium, a CD-ROM, DVD, any other optical medium, punch cards, paper

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0120

tape, any other physical medium with patterns of holes, a Random Access Memory (RAM), a programmable read only memory (PROM), an erasable programmable read only memory (EPROM), a flash electrically erasable programmable read only memory (FLASH-EEPROM), any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0035] Various forms of computer readable media in this disclosure may be involved in carrying data (e.g. sequences of instructions) to a processor. For example, data may be (i) delivered from RAM to a processor; (ii) carried over a wireless transmission medium; (iii) formatted and/or transmitted according to numerous formats, standards or protocols, such as Ethernet (or IEEE 802.3), Bluetooth, and Transmission Control Protocol/Internet Protocol ("TCP/IP"), Time Division Multiple Access (TOMA), Code Division Multiple Access (CDMA), and 2G/3G; and/or (iv) encrypted to ensure privacy or prevent fraud in any of a variety of ways well known in the art.

[0036] Thus in some embodiments in this disclosure a description of a process may be a description of a non-transitory computer-readable medium storing a program for performing the process. The computer readable medium may store (in any appropriate format) those program elements which are appropriate to perform the method.

[0037] In an embodiment, a server computer, network element orcentralized authority may not be necessary or desirable. For example, an embodiment may be practiced on one or more devices without a central authority. In such an embodiment, any functions described herein as performed by the server computer or data described as stored on the server computer may instead be performed by or stored on one or more such devices.

[0038] In interpreting the present application (which includes the claims), one of ordinary skill in the art shall refer to the prosecution history of the present application, but not to the prosecution history of any other patent or patent application, regardless of whether there are other patent applications that are considered related to the present application, and regardless of whether there are other patent applications that share a claim of priority with the present application.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0121

[0039] Numerous embodiments are described in the present application, and are presented for illustrative purposes only. The described embodiments are not, and are not intended to be, limiting in any sense. One of ordinary skill in the art will recognize that the disclosed embodiment(s) may be practiced with various modifications and alterations, such as structural, logical, software, and electrical modifications. Although particular features of the disclosed invention(s) may be described with reference to one or more particular embodiments and/or drawings, it should be understood that such features are not limited to usage in the one or more particular embodiments or drawings with reference to which they are described, unless expressly specified otherwise.

[0040] The present disclosure is not a literal description of all embodiments of the invention(s). Also, the present disclosure is not a listing of features of the invention(s) which must be present in all embodiments. A description of an embodiment with several components or features does not imply that all or even any of such components/features are required. On the contrary, a variety of optional components are described to illustrate the wide variety of possible embodiments of the present invention(s).

[0041] The following description has been presented for purposes of illustration and description and is not intended to be exhaustive or to limit the embodiments to the precise form disclosed. Many modifications and variations are possible in light of the teachings disclosed herein.

[0042] The embodiments were chosen and described to explain principles of operation and their practical applications. However, the scope of the invention is to be defined by the claims.

[0043] The major technical reason for inflexibility in network infrastructure has been related mostly to the proprietary hardware and rigid software architecture in the existing network elements (routers, switches, gateways, cellular base-stations, WiFi access points, etc.). With recent advancements in semiconductor device manufacturing, there is no longer a cost barrier to add computing power to many electronics in the market including networking infrastructure. The addition of such compute capabilities would open up new opportunities for programmable platforms in the network and ultimately creating more flexible network architecture

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0122

and business models.

[0044] The Distributed Software Defined Network (dSDN) disclosed herein is an end-to-end architecture that enables secure and flexible programmability across a network with full lifecycle management of services and applications (fxApp). The dSDN also harmonizes fxApp deployment across the network independent of the hardware vendor. As a result, the dSDN simplifies the network deployment lifecycle from concept to design to implementation to decommissioning. In this disclosure, the dSDN is applied to the wireless networks as an exemplary embodiment. However, the dSDN is not limited to wireless networks embodiments and it could be applied to many other network types including enterprise, wireline service providers, data centers, and Internet of Things (IoT).

[0045] Figure 1 demonstrates the network architecture of a 4G system 100. (In alternative embodiments, the system and method disclosed may also be implemented in a 2G/3G network as well). In a cellular network, the base station or BS (denoted as BTS in 2G, NodeB or NB in 3G, and eNodeB or eNB in 4G systems) is responsible for termination of a wireless link (air-interface) to the mobile device (also known as the User Equipment (UE)). The BS thentransports the user data to the core network and communicates control-signaling messages with the core network. In a typical cellular network, there are thousands of BS's covering nations. Each BS covers an area known as a cell. As the demand for mobile Internet expands, the mobile network operator (MNO) needs to deploy smaller cells so it can reuse the frequency more often.

[0046] A BS is made up of the following main functional blocks: 1) radio frequency (RF) front end; 2) clock; 3) baseband; 4) power manager; and 5) central processing unit (CPU). Some BS's also have a dedicated packet processor (PP) to accelerate packet processing in hardware. BS vendors may also use custom or merchant System-on-a-Chip (SoC) or Multi-Chip-Packaging (MCP) solutions to combine various functions into a single chip (e.g., PP, CPU, and baseband).

[0047] Even though there has been attempts to open up various interfaces in the BS (such as the Open BS Architecture Initiative that defines interfaces between these functional modules in the BS), there has been very little to no efforts to unify the

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0123

programmability of the BS itself. As a result, the MNOs suffer from the following difficulties. First, the core network Internet edge (CNIE), Packet Data Gateway (PDG) and the surrounding functions are becoming extremely complicated, non scalable, and expensive. Second, each MNO needs certain customization and feature sets. Currently, they depend on their vendors for these customizations, which could cost the MNO tremendously both financially and in regards to time-to-market (TTM); hence, hindering innovation. In addition, usually these features are put into standards or in the vendor's feature set, thus eliminating the MNO differentiation against the other MNOs. Third, if the MNO is multi-vendor in their CNIE, coordinating all the vendors to implement the same features is usually a difficult and a time-consuming effort. As a result, more add-on network appliances are introduced into the CNIE which adds to the network management complexity. Fourth, as a BS becomes Internet Protocol (IP) based, there is more visibility into the user traffic types and new innovative opportunities are missed for creative traffic shaping features, backhaul bandwidth optimization, prioritization at radio edge, power management algorithms, etc. Fifth, many features implemented in CNIE present suboptimal performance. For example, the filtering enforcement at CNIE is fundamentally inefficient since the packets have to travel all the way through the expensive air interface, the backhaul, various other core network elements and transport networks to get to the CNIE. Sixth, the Radio Access Network (RAN) deployments are a very expensive endeavor for MNOs. The current rigid BS designs limit innovation and force the MNOs to undergo major upgrades every few years.

[0048] The problems stated above apply to many different types of networks even though here they are presented in cellular networks as a focus. In the present disclosure, the dSDN exhibits a new paradigm in the software programmability of networks that would address the problems above and enable many more advanced features.

[0049] The dSDN may be located at a network element (or system) 200 which is shown in detail in Figure 2 or in alternative embodiments the functions of the dSDN may be divided among a plurality of network elements (or systems) which are similar in structure to network element 200. Each network element 200 may comprise one or

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0124

more system control logic 202 coupled with at least one or a plurality of processor(s) 204, system memory 206, a network interface 208 (including a transceiver 208a), and input/output (1/0) devices 210. The processor(s) 204 may include one or more single core or multi-core processors. The processor(s) 204 may include any combination of general-purpose processors and dedicated processors (e.g., graphics processors, application processors, baseband processors, etc.). System control logic 202 for one embodiment may include any suitable interface controllers to provide for any suitable interface to at least one of the processor(s) 204 and/or to any suitable device or component in a packet network in communication with system control logic 202. System control logic 202 for one embodiment may include one or more memory controller(s) to provide an interface to system memory 206. System memory 206 may be used to load and store data and/or instructions, for example, for network element 200. System memory 206 for one embodiment may include any suitable volatile memory, such as suitable dynamic random access memory (DRAM), for example. System memory 206 may also include non-volatile memory including one or more tangible, non-transitory computer-readable media used to store data and/or instructions, for example, such as the embodiments described herein with regard to the **dSDN.** The non-volatile memory may include flash memory, for example, and/or may include any suitable non-volatile storage device(s), such as one or more hard disk drive(s) (HDD(s)), one or more compact disk (CD) drive(s), and/or one or more digital versatile disk (DVD) drive(s). The system memory 206 may include a storage resource physically part of a device on which the network element 200 is installed or it may be accessible by, but not necessarily a part of, the device. For example, the system memory 206 may be accessed over a network via the network interface 208 and/or over Input/Output (1/0) devices 210. Network interface 208 may include a transceiver 208a to provide a radio interface for network element 200 to communicate over one or more network(s) and/or with any other suitable device. Network interface 208 may include any suitable hardware and/or firmware. The network interface 208 may include a plurality of antennas to provide a multiple input, multiple output radio interface. Network interface 208 for one embodiment may include, for example, a wired network adapter, a wireless network adapter, a telephone modem, and/or a wireless modem. For one

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0125

embodiment, at least one of the processor(s) 204 may be packaged together with logic for one or more controller(s) of system control logic 202. For one embodiment, at least one of the processor(s) 204 may be integrated on the same die with logic for one or more controller(s) of system control logic 202. In various embodiments, the 1/0 devices 210 may include user interfaces designed to enable user interaction with the network element or system 200, peripheral component interfaces designed to enable peripheral component interaction with the network element or system 200, and/or sensors designed to determine environmental conditions and/or location information related to the network element or system 200. In various embodiments, the peripheral component interfaces may include, but are not limited to, a non-volatile memory port, a universal serial bus (USB) port, an audio jack, and a power supply interface.

dSDN Architecture

[0050] In the dSDN architecture, the network features are virtualized and may be distributed across separate network elements that cooperate together to create an advanced, programmable, and scalable network. In an alternative embodiment, the network features may be located at a single network element. A high-level overview of a dSDN system 300 is depicted in Figure 3. The dSDN system 300 consists of a flexible network device (fxDevice) 302, a flexible cloud platform (fxCloud) 304, an application management portal (fxManager) 306, and an infrastructure application market place(fxStore) 308. The fxDevice 302 is a network device that is powered by a sandboxing operating system named flexible operating system (fxOS) 302a. The sandboxing operating system ensures each application runs as a dedicated process in sure isolation from the other applications. The fxOS 302a may be built based on an existing sandboxed operating system (OS) (e.g., Android®) by extending several aspects of such an OS with routing/networking layers and supports for operating as the wireless infrastructure. The fxDevice 302 may host several independent and securely isolated applications (on top of fxOS 302a) named here fxDeviceApp 302b.

[0051] In the dSDN system, fxDeviceApp 302b may have a sister app in the backend cloud infrastructure (i.e., flexible cloud platform 304) referenced here as fxCloudApp 304a. The fxCloudApp 304a in the cloud is paired with its fxDeviceApp 302b in the fxDevice 302. The fxCloudApp 304a and the fxDeviceApp 302b

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0126

collectively form a distributed application (dApp or fxApp). In this description, when fxDeviceApp is referenced it shall mean any application that may have software components in the fxDevice, fxCloud, or both. In general, an fxApp package may include the following software components:

fxDeviceApp binary;

fxCloudApp binary;

Manifest files; and

Signatures.

[0052] It is important to note that the fxDeviceApp 302b and fxCloudApp 304a could use any protocol to communicate with each other and there is no need for standardizing this communication allowing for ultimate freedom for the developers. This allows for the system to operate in a loosely coupled autonomous fashion allowing for asynchronous communication in a distributed fashion.

[0053] The fxApps lifecycle is managed via fxManager 306. The fxManager 306 presents a user-friendly portal to the network administrator (admin) to discover, test prior to deployment, provision, and deprovision dApps. Using fxManager 306, the admin may discover new fxApps from the fxStore 308, which presents all the tested and certified vApps as well as showing the supported fxOS version, support hardware platforms, and other information such as reviews and number of commercial deployments by all NOIT customers.

fxOS

[0054] fxDevice 302 is defined as any networking equipment embedded with a special flexible operating system (fxOS) 302a. The purpose of fxOS 302a is to enable a carrier-class programmable platform. The fx:OS 302a is configured to do, but is not limited to, the following. First, since fxOS 302a is designed for network equipment, it allows for efficient execution and running all the time. A plurality of power save modes may be implemented to ensure energy saving schemes without hindering the functionality of the fxDevice 302. Second, it enables simultaneous execution of several flexible applications (fxDeviceApp 302b) on the fxDevice 302. Third, it allows for application of a legacy compiled binary or byte-code of another language (for example, Java or Android®) in addition to fxDeviceApps 302b. Fourth, it allows

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0127

FastPath processing for data plane packets. The packet forwarding in this case separated from the control plane operations. The FastPath engine allows integration with specialized hardware accelerators or packet processors (PP). The packet forwarding may be implemented with software/kernel enhancements all in software. Another FastPath implementation may use both software/kernel enhancement and hardware packet processors (PP). Fifth, the fxOS 302a is capable of Deep Packet Inspection (DPI) and identifying all the flows and their related protocols, services and present this information in a format useful to application developers. The fxOS 302a can act on the data flows as the directed by the application developer. Sixth, applications are securely isolated from each other and to communicate with them is completely configurable by policies defined by the admin. Seventh, resource utilization of each application is policed such that an application may not be able to exceed its usage. For example, an application is pre-allocated the memory size, cache/storage space, and CPU cycles. Eighth, the applications may be signed with unique certificate security keys. The security keys and certificates may be allocated and/or signed by a certificate authority. The fx:Device 302 may be protected by validating signed applications to run or to install in the device. Ninth, it allows hot upgrade of the software and applications on the platform with little to no interruption to the operational aspect of the platform and its applications. Tenth, the fxOS 302a may boot by using the secure code from the network servers (cloud-booting). This would ensure tamper-proof software foundation for the fxDevice 302. Eleventh, there may be a secure and programmable firewall in the fxOS 302a. This embedded firewall protects against attacks to the fxDevice 302 and network attached to the network. The fx:OS 302a firewall may be configured by the developer and it does allow replacing of attack detection algorithms (e.g., man-in-the-middle detection algorithm). The fxOS 302a may allow secure caching of user data, web pages, and media files (video and audio) in the fxDevice 302. The secure connections to the cloud (fxCloud 304) can enable sharing and control of the caching between the fxCloud 304, fxDevice 302, or the rest of the network. Twelfth, the fxOS 302a can support multi-threading and multi-core CPUs. This feature may dynamically distribute the threads to various CPU core without developers assistance.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0128

[0055] The fxOS 302a allows secure connection to the fxCloud 304. This secure connection allows communication between fxDeviceApps 302b in the fxDevice 302 and fxCloudApp 304a in the fxCloud 304. The fxOS 302a controls the access of applications to platform APIs. An application is categorized into an Access Level (AL) based on which the fxOS 302a decides that the application does not exceed its AL and access APIs that it is not authorized. The fxCloud 304 and the fxOS 302a create a virtual fabric (fxVF) for messaging between applications. The actual message can travel directly between two fxDevices or it may traverse the fxCloud 304. The messaging complexity may be abstract for the developer.

[0056] Figure 4 demonstrates an examplary view of the software layers in the fxOS 302a. Each layer of this software stack is modified and redefined to optimize for embedded networking equipment of wireless and wireline nature. Starting from the bottom of the stack, the kernel also connects to the hardware resources via the device drivers. Examples of such resources are wireless baseband SoC402, hardware packet processor 404, Wide Area Network (WAN) uplink interface 406, cache and storage 408, and multi-core CPU 410. The OS kernel 412 is enhanced with the FastPath function 414 to accelerate packet processing (by help of hardware features such as Intel's Data Plane Development Kit (DPDK) or pure software enhancements) and other related data path functions. The enhanced kernel with FastPath 412,414 would enable visibility into the data flows, sniffing the Non-Access Stratum (NAS) signaling (which is the signaling between the UE and the Core Network and the RAN is supposed to be transparent to), policy enforcements, regular packet forwarding functions, collection offlow statistics into Flow Information Base (FIB), and encryption/decryption functions.

[0057] Above the kernel, there are several tools 416, native daemons 418, native libraries 420 and Hardware Abstraction Layers (HAL) 422. These tools abstract out the hardware dependencies for the upper layers and programmers. The Runtime 424 is an embedded virtual machine capable of securely isolating and executing the applications. System Services 426 are a set of services always running and available to the developers (e.g., timing and messaging services).

[0058] fx-Framework Library 428 consists of several Frameworks. The new

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0129

frameworks are the library extensions introduced explicitly for the dSDN framework. Each of these frameworks adds a set of methods of (functions) and data structures for the following examples:

Wireless Manager 428a: manipulation, monitoring, and configuration of the wireless interfaces;

System Manager 428b: configuration, provisioning, and changing major settings of the fxDevice 302;

Flow Controller 428c: allows for manipulation of data path, tunneling, switching, and optionally coordinate this activity with fxCloud 304;

Energy Management 428d: enables fine-grained control over various elements in the fx:Device 302 that could be moved into various power stares, for example: ON, OFF, SLEEP, SUSPEND;

Big Data Manager 428e: manipulation, processing, and organization ofdata collected from various elements. This framework could enable close coordination between the fx:Device 302 and thefxCloud 304;

Content Manager 428f: for data sharing (database sharing) between the applications within the fxDevice 302 or the fxCloud 304 and between applications in fxDevice 302 and fxCloud 304. The Content Manager 428f may use URI to address the contents to be shared. The Content Manager 428f allows access to the application data according to the policy settings and permissions;

Policy Manager 428g:_responsible for execution and implementation of polices set up by the admin. The application developers could use this framework to perform queries about the permissions, limitations, and rules;

Security Manager 428h: allows for access to the security protocol libraries and rewriting some of security algorithms such as the man-in-the-middle detection algorithm;

General Framework: general compute and logic building that may be

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0130

inherited from existing general OS frameworks.

[0059] Using the frameworks, there are potentially at least two types of application types that are possible. First, the System Native Applications 430 that are provided as the initial application load into the platform. These applications could be used by other applications by links or API. Second, the fxDeviceApps 302b that are written by third party developers or the NOIT.

[0060] Figure 5 further details the FastPath 414 architecture. In this approach, the FastPath 414 is divided into the Virtual Switch (data-plane) 502 and the Switch Controller (control-plane) 504. Such separation makes possible flexible deployment wherein the function of the Switch Controller 504 could move to the fxCloud 304 by an admin decision or by the dSDN system intelligence for example in the case of failure, redundancy, stateful reboot, or Hot Upgrade. The Virtual Switch 502 may include several functions and may interact with the Switch Controller 504 in the following exemplary ways:

Firewall 502a: implements the policies given to it by the firewall logic 504a in the fxOS Daemons.

Flow/NAS Discover 502b: continuously monitors data packets and discovers new flows and NAS signaling. It would then report it to the DPI Logic 504b where the flow intelligence resides and matching to meta-data happens. DPI Logic 504b in turn updates the Flow Information Base Information (FIB) 502c of the Virtual Switch 502.

FastPath & DPI Processing 502d: forwards, routes packets, and performs DPI functions as instructed by the FIB 502c. It also collects statistics of packets and reports them to the Monitoring Logic 504c.

Policy Enforcement 502e: this function enforces various policies on the traffic such as QoS marking according to the policies received from the Policy Logic 504d.

Shaper 502f: this function could shape (queue) the traffic according to the available bandwidth on the uplink or WAN interface.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0131

Virtual Fabric (fxVF) 504e: enables transparent switching of application and system messages of a dApp between the fxCloud 304 and the fxDevice 302 and between different dApps. The fxVF 504e is an abstraction layer that hides most of the complexities of messaging from the developers. The fx:VF 504e uses the policies and firewall rules to enforce security and policies.

[0061] The fxOS 302a is mostly used for networking applications. There are some potential use cases where there is a need for a general purpose OS to run the legacy software applications with no need of recompilation. An example of a double OS situation may be where a digital signage (using a general purpose OS such as Android®) in a shopping mall also acts as a smart small-cell (using the fxOS 302a). In such a solution, any existing Android®application (for example) could be loaded in the fx:Device 302 and in parallel networking applications (fxDeviceApps 302b) could run on the same fxDevice302.

[0062] Figure 6 presents a potential software layer of the fxOS 302a co-existing alongside a General Purpose OS 602. Another implementation may be a standalone fxOS without the company of another general OS. It is important to note that the shown co-existence allows reuse of the kernel so there would be no need to re-port the drivers of hardware resources twice.

fxCloud

[0063] The fxCloud 304 is an integral part of the dSDN framework. The fxCloud 304 connects to various parts of the fxDevice 302 to form a single virtual view of the system. Examples of the main fx:Cloud 304 responsibilities include the following. First, it interacts and manages all applications, firmware, and fxOS 302a in the fxDevice 302. Second, it secures isolation environments for distributed apps. Third, it directs the Virtual Machine lifecycle management including load-linked VM creations and destruction (Cloud Breathing). Fourth, the fxCloud 304 executes instructions defined by the fxManager 306. Fifth, it implements uniform policy (including security) distribution and execution across dSDN 300 (including the fxDevice 302, fxDeviceApps 302b, fxCloudApps 304a, anddApps). Sixth, Plug-and-Play commissioning of thefxDevice 302 introduction

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0132

and network expansion. Seventh, Distributed Resource Service (dRS) by which distributed applications can seamlessly communicate and message to each other using published APIs or proprietary methods within the pre-defined policies. Eighth, security violation discovery and alert system. Ninth, translation and/or exposure of Resources APIs (RAPI) to the systems outside the dSDN 300 within pre-defined policies. Tenth, fxCloud 304 in coordination with the fxDevice 302 creates a Distributed Notification Service (dNS) across the dSDN 300 by which the applications could be notified of an event to wakeup and/or respond to the event. Eleventh, fxCloud 304 in coordination with thefxDevice 302 creates a Distributed Content Management Service(dCMP)which provides a virtualized and distributed database system for applications to share application-specific or system-specific information with the pre-defined policies. Twelfth, the Switch Controller 504 function in the fxCloud 304 could take control of the controller function of the fxOS 302a in the fxDevice 302. This would allow controller switching from local to remote and back. Examples of such a configuration is a temporary reason where the fxOS Switch Controller 504 may not be available during reboot or software upgrade.

[0064] In the discussion below, these services and features are expanded upon.

fxDeviceApp-0 & fxCloudApp-0

[0065] The fxDevice 302 and fxCloud 304 have embedded System Applications categorized as fxDeviceApp-0 and fxCloudApp-0, respectively. ThefxDeviceApp-0 is responsible for all default and core functional capabilities of the fxDevice 302 as listed below:

Firmware of thefxDevice;

Default communication protocols (e.g. LTE S1-AP, baseband software, security protocols, GTP, DNS, tunneling protocols);

Message communication protocol between the fxDevice and fxCloud;

Initial fxDevice settings;

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0133

Boot Loader;

fxDevice X.509 digital certificate;

OS and Kernel upgrades;

Upgrade and commissioning agents for the fxDeviceApp-0 upgrades and related procedures;

Establishing and initiating the message communication protocol between the fxDevice and fxCloud;

General fxDeviceApp install and upgrades;

Access to fxDevice hardware and software information.

The fxCloudApp-0 is responsible for all default and core functional capabilities of the fx:Cloud 304 such as:

Upgrade and commissioning agents for the fxCloudApp-0 upgrades and related procedures;

Hypervisor upgrades;

Virtual Machine lifecycle management (e.g.creation, suspension, destruction);

fxCloud X.509 digital certificate management;

OS and kernel upgrades;

General fxCloudApp install and upgrades;

Default communication protocols (e.g. security protocols, DNS, tunneling protocols).

Each software component can be managed and updated individually. To simplify the management of these software components, each software components can be sub-numbered such as fxDeviceApp-0.1, fxDeviceApp-0.2, fxCloudApp-0.1, fxCloudApp-0.2, etc.

fxManager

[0066] The network administrator may use the fxManager 306 to control much of the dSDN system 300. The fxManager portal could use HTML5, for example, as

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0134

a frontend technology. The major functions of the fx:Manager 306 may include the following.

fxDevice Management:

Commissioning (plug-and-play), de-commissioning, firmware/OS upgrading/downgrading;

Suspend/resume functions;

Unification of zones and partitions for example in case of network merger;

Detailed management of fxDevices (remote secure login) where the administrator may zoom into a particular or a group offxDevices to monitor and setmanagement alarms;

Security policy setting.

Application lifecycle management: fxManager 306 may work, coordinate, and/or perform these tasks in conjunctions with fxCloud 304 for:

Provisioning, de-provisioning, upgrade/downgrade applications. Some applications may only have a fxDevice component (Device Centric Apps), may only have fxCloud component (Cloud-Centric Apps), or may have both fxDevice and fxCloud components (Hybrid Apps);

Creating a Test Network Environment (TNE) where the administrator can test certain configuration and applications without impacting the production networks;

Uploading applications developed by the NOIT or their partners into the Application Repository (AR). The AR is placeholder where the application packages are downloaded and stored for testing and verification prior to installation into the live dSDN system. An application could have several components (e.g. fxDeviceApp binaries, fxCloudApp signatures, manifest files, signatures) bundled together to form the Distributed Application Package (dAP);

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0135

Browsing through the certified applications in the fxStore and downloading them to theAR;

Application license accounting services by which a secure and accurate accounting of the installed applications are performed regularly. This inventory management function allows the administrator and the 3rd party ISV toaccount for live and active instants of the software in the network;

Application review tool where the network admin could use to review feedbacks and comments of other NOIT (maybe anonymous reviews) and could input his/her reviews for others (maybe anonymous reviews).

fxCloud Management:

Zoning and partitioning the network for administrative purposes. Zones and partitions could optionally put into different Virtual Machine (VM);

VM load management and Cloud Breathing policies;

The configuration conflict resolutions. Some configurations applied to the resources may conflict with each other (e.g., one firewall rule may state drop Netflix packets explicitly and another rule may state allowing Netflix explicitly). The fxCloud resource manager could monitor these configurations and discover the conflicts and signal those to the networkadministrator.

Security Management:

Security policy setting;

Inter-App communicationsetting;

Security of communication between fxDevice and fxCloud;

API Access Level (AL) definitions and (re-)assignment to the applications (permissions);

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0136

Risk Analysis of the applications;

Certificate and signature verification of the applications;

Content sharing between applications and associated policies.

[0067] An fxManager Portal per the above description may have an "Account" button which can be used to create admin users and their privileges. The fxManager Portal could also include a geographical map illustrating locations of fxDevices and their status. The fx:Manager 306 functions could be integrated into other network management tools. The fxManager 306 could potentially build the inventory of the network devices (fxDevices) showing all the available devices and statistics. As described above, the network administrator could optionally procure the applications a third party hosted application store (fxStore). The fxStore 308 could potentially categorize the applications several groups (as shown in Figure 7 which is an example view of the fxStore portal). The fxStore 308 could also show the reviews of the advertised applications. In order to maintain quality, the third party fxStore 308 may perform rigorous testing of the applications and accompany those results with the advertised application.

fxSDK

[0068] The flexible Software Development Kit (fxSDK) may include a development environment and other tools to facilitate development, testing, debugging and verifications of applications for dSDN environments. The fxSDK is unique in several aspects exemplified as follows. First, since the dSDN applications may have a device component (fxDevice 302) and cloud component (fxCloud 304), the fxSDK allows the developer to develop the components together to simplify the development and testing. Second, the dRMS offers various system level APIs with remote accessibility. In other words, the APIs in the fxDevice 302 could be accessed via the applications in the fxCloud (fxCloudApps) and vice versa. The fxSDK simplifies the usage system created APIs and application created APIs. Third, it is essential for the dSDN to ensure quality and security of applications in the network. The fxSDK may be accompanied with an extensive Risk Analysis Utility (RAU) which verifies all the APIs that are being used by the application and based on its potential danger to the

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0137

network, it would show the risk analysis and may make specific suggestions to reduce the risk. Once completed, the risk analysis result is included in the dAP for posting to the fxStore and consumed by the fxManager user (the admin). Fourth, the fxSDK would also ensure that the applications do not exceed their planned memory and resource usage. Fifth, the Integrated Development Environment (IDE) may provide various tools to actively demonstrate errors, defined APIs, security dangers, and excessive resource usage. Sixth, the IDE could also provide a flexible Simulator (fxSimulator) for the developer test the application in a network setup in presence of other network elements.

Deployment Examples

[0069] The dSDN 300 offers flexible deployment options. In this section, a few exemplary embodiments are presented.

[0070] Figure 8 demonstrates a shared cloud deployment where several NOITs (802,804,806) share an fxCloud 808. In order to ensure complete isolations, each NOIT may be assigned a separate Virtual Machine (VM) 802a, 804a, 806a. A good example of this deployment option may be when multiple service providers outsource their fxCloud deployment to a third party vendor that offers a SaaS/PaaS solution. It is important to note that in this option, each NOIT should have its own secure instance of the whole system. This option could potentially create a full outsourcing model for network operations including possibilities of seamless merging of networks of two or more NOITs.

[0071] Another deployment option may be a Zoned Deployment as shown in Figure 9. In this option, a large NOIT may partition their network into a plurality of zones (Zone #1, Zone#2,... Zone#n) for administrative, software compatibility grouping, or other reason. For example, it is possible that in a large network deployment, there would be different versions of fxOS with varying capabilities in the network (the older hardware may not support newer fxOS version). In this case, the service provider may decide to group the fxDevices per fxOS versions for management simplifications.

[0072] Figure 10 is another embodiment deployed in a Programmable & Virtualized Cellular Network. In this example, the dSDN is applied to the cellular

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0138

networks (e.g., LTE). As shown, the fxDevice 1002 here is the LTE eNB (4G base-station) and the fxCloud 1004 hosts the LTE core network (EPC) functions such as MME, S-GW, and P-GW in separate VMs. Within each EPC function, new features could be added using the dSDN framework. For example, a new application to optimize signaling for Machine-2-Machine (M2M) devices could be loaded into the fxDevice 1002 and the MME function in the fxCloud 1004.

PRINCIPAL PROCEDURES & SERVICES

[0073] The dSDN framework could offer extensive services and capabilities that would simplify the programmability of the network. In this section below, a set of main procedures and services are presented as further exemplary embodiments.

fxDevice Commissioning Provisioning

[0074] Figure 11 illustrates a signaling flow chart fora fxDevice Plug-and-Play Commissioning Procedure. This procedure enablesfxDevice commissioning and initial setup to provide a seamless and Plug-and-Play (PnP) deployment. The fxDevice relies on the fxDeviceApp-0 and the fxCloudApp-0 to connect to the backend management system and perform the commissioning procedure. In step 1101, the fxDevice 1120 tries to discover the fxManager 1130 using the fxManager Fully Qualified Domain Name (FQDN). The fxDevice 1120 is configured to always look for the same FQDN. If the NOIT has a private cloud deployment, the DNS server in the NOIT will be configured to point to its own fxManager IP address. If the cloud is shared, the NOIT will configure their DNS server to point to the shared cloud IP address. In step 1102, the fxDevice 1120 receives the IP address of the fxManager 1130. In step 1103, the mutual authentication may be performed where the fxDevice 1120 authenticates the fxManager 1130 and vice versa. In step 1104, a secure tunnel may be created between the fxDevice 1120 and the fxManager 1130. In step 1105, the fxDevice 1120 may optionally request for the latest firmware/OS version. In this phase, the fxDevice 1120 informs the fxManager 1130 of its hardware information (e.g. vendor, model number, CPU model/speed, memory/cache size) and software information (e.g., current firmware version, current OS version, fxDeviceApp-0 version package statingversionsoffxDeviceApp-0 components). In step 1106, the fxManager 1120 may send the latest firmware/OS version (or point

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0139

the fxDevice to the right URL to download the latest load). In step 1107, if a new firmware/OS load is downloaded and verified, the fxDevice 1120 installs it. In step 1108, the fxDevice 1120 requests the default settings and the default application package (the initial applications that the NOIT admin would like to install in fxDevice 1120 prior to other applications). In step 1109, the fxManager 1130 prepares the fxCloud 1140 for the new fxDevice 1120 integration into the network. In step 1110, once completed, the fxCloud 1140 informs the fxManager 1130 of its readiness. In step 1111, the fxManager 1120 sends all the default settings and the default application package to thefxDevice 1120. In step 1112, the fxDevice 1120 then sets up a secure connection to the fxCloud 1140 (more precisely the fxCloudApp-0). In step 1113, a commissioning completion message is set from the fxCloud 1140 to the fxManager 1130. In step 1114, the fxManager 1130 signal to the fxDevice 1120 the completion of the commissioning and starts th regular operation of the fxDevice 1120. At this phase, fxManager 1130 can optionally modify the settings in the security connections between fxDeviceApp-0 and fx-CloudApp-0 and between fxDeviceApp-0 and fxManager 1130.

Application Provisioning Procedure

[0075] The Application Provisioning Procedure enables provisioning and de provisioning of applications across the dSDN system. For simplicity, the term provisioning in this disclosure is used to present all similar procedures of provisioning and de-provisioning. The application provisioning procedures are usually triggered by the network administrator and are orchestrated by the fxManager, which works in collaboration with the fxDeviceApp-0 and fxCloudApp-0.

[0076] Figure 12 shows an example procedure for Application (dApp) Provisioning. In step 1201, the network admin identifies the applications that would need to be installed in the dSDN system. These applications are already downloaded (from the fxStore or other sources) into the Application Repository (AR). Prior to installation in the live network, the authorized network admin 1220 has probably performed rigorous testing of the application in the test network. In step 1202, the fxManager 1230 performs a detailed analysis on the potential risks that this application may

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0140

create on various parts of the network and other applications. The analysis tool uses the secure manifest file and other information to discover, for example, all the APIs used by the application, the required access-level authorization level for each of those APIs, requirements on the fxVF (inter-app, intra-app communication needs), and usage of platform resources including conflict discovery and resolutions. In step 1203, the admin defines the deployment scope that the application would be applied. Examples of deployment scopes could be a given zip code, a city, an administrative domain (zone), a shopping mall, an office building, or manual handpicking of sites or fxDevices 1240. The fxManager 1230verifies compatibility ofthe OS version and resouces requirements of the application and the fxDevices 1240. If the fxManager 1230 discovers incompatibility, it informs the admin and request adjustment to the deployment scope or proposes upgrading incompatible fxDevice if possible. In step 1204, the admin 1220 initiates the installation process. In step 1205, the fxManager 1230 unpacks the dApp and if there is fxDeviceApp component, it would prepare it for submission to the fxDeviceApp-0 of fxDevices in the deployment scope. In step 1206, installation commands are sent to the fxDeviceApp-0s in a secure connection already set up as part of the fxDevice Commissioning. In step 1207, the fxDevice 1240 verifies the fxDeviceApp package and performs necessary integrity checks of the received software package. Once verified, the fxDeviceApp is installed and re verified. In step 1208, the completion of successful install is sent to the fxManager 1230 by the fxDevice 1240. In step 1209, the fxManager 1230 prepares the fxCloudApp component of the dApp (if any). In step 1210, installation commands are sent to the fxCloudApp-0s in a existing secure connection. In step 1211, the fxCloud 1250 verifies the fxCloudApp package and performs necessary integrity checks of the received software package. Once verified, the fxCloudApp is installed and re-verified. In step 1212, the completion of successful install is sent to the fxManager 1230 by the fxCloud 1250. In step 1213, the fxManager 1230 informsthe admin 1220 of the successful installation process.

Hot Upgrade Procedure

[0077] The Hot Upgrade refers to a procedure by which the software upgrades

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0141

on the system have no or minimal implication on the functionalities offered by the software component subject to the upgrade process. Herein are presented two types of software upgrades. For simplicity, the use of the term upgraded in this disclosure represents both upgrade and downgrade since they both use identical processes. First, upgrade of applications on the fxDevice (i.e., fxDeviceApp) and fxCloud (i.e., fxCloudApp). Second, upgrade of core functions in the fxDevice (aka fxDeviceApp-0) or fxCloud (aka fxCloudApp-0).

[0078] The upgrade procedures for fxDevice and fxCloud are similar. Here, the focus is on the fx:Device upgrade process since it is technically more challenging due to stricter resource limitations in the fx:Device.

General Application Upgrade

[0079] The applications could be upgraded automatically or manually by the network admin. In either case, the installation of the upgrade follows a similar procedure as the Application Provisioning Procedure. Figure 13 presents the state of the machine for the application upgrade process. In state 1301, the old version of the application is still running. In Load state 1302: the new version of the application is downloaded in the local storage while the old version is still running. In Install state 1303: the new version is installed (in the memory) while the old version is still running. In Run state 1304: the new version is activated taking charge of all the related data and application state and the old version is deactivated while still remaining in the local storage. In Commit state 1305: the old version is fully removed after ensuring the new version is running properly as expected.

fxDeviceApp-0 Upgrade

[0080] The fxDeviceApp-0 is a collection of software components that perform the core functionalities of the fx:Device. As a result, this hot upgrade is quite challenging. The most important thing is to ensure that major functionalities of the fxDevice (such as packet forwarding) remain

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0142

intact during the upgrade process. The fxDeviceApp-0 upgrades may be categorized as:

- No OS Kernel upgrade; or
- OS Kernel upgrade.

No Kernel Upgrade

[0081] As described earlier, the packet forwarding function could be broken into the Virtual Switch and the Switch Controller. In order to maintain, the packet forwarding function, the Switch Controller function could be performed by another device or a second processor in the same device while the fxDevice is being upgraded. Figures 14A-14C demonstrate a few examples of distributed Switch Controller function during the upgrade. In some cases, it may be possible for the fxCloud to take over the Switch Controller function (Figure 14A), or a neighboring fxDevice (Figure 14B), or second CPU in the same fxDevice (Figure 14C). The actual software upgrade procedure would be similar to Application Provisioning Procedure and the state machines are the same as the one shown in Figure 13.

Kernel Upgrade

[0082] In this case, the packet forwarding process is mostly unavailable. Therefore, one potential solution might be for the fxDevice to redirect the entire bit stream to a neighboring fxDevice. In other words, the fxDevice would have just a bare minimum routing function working and the rest of the function would be performed by a neighboring fxDevice.

[0083] All the other upgrade procedures and state machines are similar to the ones performed for the No Kernel Upgrade process.

Virtual Fabric Service

[0084] The Virtual Fabric (fxVF) provides an abstraction layer for application to communicate with each other whether they are in the fxDevice or fxCloud. Various frameworks and services may use the fx:VF service. Figure 15 illustrates fxVF 1502a, 1503a, 1504a in the fxDevices 1502 and 1503 and fxCloud 1504. The fxVF 1502a,

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0143

1503a, 1504a may use Extensible Messaging and Presence Protocol (XMPP) for this messaging as the default protocol. It is important that the developers may use their own communication protocols between the fxDeviceApps 1502b, 1503b and the associated fxCloudApp 1504b. fxVF 1502a, 1503a, 1504a provide a secure routing mechanism.

Intra-App Messaging

[0085] This is an example of where one application in the fxDevice communicates with the same instance of the app in another instance of the same application in another fxDevice. The actual messages could go directly between the fxDevice or via the fxCloud. As an example, in the case of mobile networks, this messaging could be used to transfer user specific context from one eNB to another as a user hands off to a new eNB. In LTE, Private Messages on X2 interface could be used for messaging between the eNBs that act as the fxDevices.

Inter-App Messaging

[0086] In some cases, different applications may need to communicate with each other via their published APIs. In this case, the security policies set up by the network administrator determines which applications could communicate with each other for what purpose. The fxVF follows the security policies determined by the network administratorfor inter-application communications.

distributed Resources Service (dRS)

[0087] The resources could be platform resources or APIs offered by the applications. The fxCloud and fx:Device resources could include (as examples):

Firewall;
Data and statistics;
Storage (usually abundant at fxCloud);
Compute (usually abundant at fxCloud);
Load evaluator;
General settings;

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0144

Routing engine (usually relevant to the fxDevice);

Virtual Machine master controller (usually relevant to the fxCloud);

Power controller (usually relevant to the fxDevice); and

Wireless engine (usually relevant to the fxDevice).

[0088] An app (fxDeviceApp or fxCloudApp) could expose APIs to be used by the other apps. The inter-app communication is enabled by the fxVF where the policy and security provisions are enforced. The APIs exposed may be RESTful (representational state transfer) and could travel across physical network elements. Since the fx:Managerdefines fxVF policies, the administrator could ultimately specify which APIs between which apps could communicate with each other. Figure 16 demonstrates the logical interfaces between the platform resources and applications. As shown in Figure 16, the distributed Resources Service (dRS) 1602, 1603 comprises software agents residing in the fxDevice 1604 and the fxCloud 1606. These dRS agents manage the access to the platform resolutions (including potential configuration conflict resolution) as well as facilitating inter-app APIs. The dRS 1602, 1603 allows or disallows access to platform resource or inter-app communication according to the policies defined by the network administrator using the fx:Manager. In summary, the dRS 1602, 1603 provides services to application developers for:

Exposing APIs to other applications;

Configuring and managing platform resource;

Policy enforcement and authorization of applications access to platform resource and otherapp's APIs; and

Policy conflict resolution.

distributed Content Service (dCS)

[0089] The distributed Content Service (dCS) allows the developers to seamlessly store and share the contents generated by one application with other applications and its associated application in the cloud. The dCS simplifies access to the data and brings in storage virtualization to the applications. In other words, the developers no longer would need to know where the data is actually stored (in

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0145

the cloud or on the device) and would be able to access them easily. The dCS implementation may use Virtual Fabric (fxVF) and distributed Resources Service (dRS).

distributed Notification Service (dNS)

[0090] The distributed Notification Service (dNS) is another potential tool for developers that could wake or ping an application when a particular event has occurred. For example, a load monitoring application could be notified when the CPU load on a particular fxDevice (or a target area) exceeds a certain threshold. In turn, such an exemplary application could make smart decisions on reducing the load on the CPU by forcing handoffs of users to neighboring cells (i.e. fxDevices). The dCS implementation may use Virtual Fabric (fxVF), distributed Resources Service (dRS) and distributed Content Service (dCS).

Cloud Breathing Procedures

[0091] In general, resources (compute, storage) at the cloud are more abundant. However, software licensing costs and other limitation may require smart management of resources. The Cloud Breathing, here, is defined as a mechanism where the cloud resources are automatically expanded as load increases on the system or reduced as the load decreases. This creates an automatic elasticity in the cloud dimensions. Figures 17A and 17B demonstrate procedures for the cloud breathing. Figure 17A is expansion and Figure 17B is reduction. The Load Controller in the dRS 1602, 1603 could be used to monitor the load (step 1702) on the existing VMs (e.g. CPU or memory utilization loads). The VM Master of the fxCloudApp-0 1608 could effectively act on the decisions made by the dRS's Load Controller. Again, the network administrator via the fx:Managerdefines the policies and thresholds for such decisions.

Security Aspects

[0092] The security aspects of this solution are of outmost importance in order to ensure quality in the productnetworks.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0146

Identity

[0093] To create a security model, the dSDN network elements should have unique identities. The following identities may be defined:

fxDevice Identity (DID): this identity is unique globally and may be allocated at manufacturing and included in the X.509 digital certificate of the fxDevice;

fxCloud Identity (CID): this identity is unique to the NOIT domain and may be allocated by the network administrator and included in the X.509 digital certificate of the fxCloud;

fxDeviceApp Identity (DAID): this identity may be globally unique and may be allocated by the fxSDK at the point of code creation by a globally accessible server;

fxCloudApp Identity (CAID): this identity may be globally unique and may be allocated by the fxSDK at the point of code creation by a globally accessible server;

dApp Identity (AID): This identity may be globally unique and may be constructed by concatenation of DAID and CAID together;

Application Developer Identity (ADID): this is a globally unique identity allocated by a globally accessible server and may be included in the developer's X.509 digital certificate and is used to sign the final developed applications.

[0094] For addressability purposes, the identities may be presented in a URI format using the FQDN of the NOIT.

Access Level

[0095] The applications are given different access levels. The AL is used by the dRS to determine which APIs or class of APIs an application can access (e.g., a fxDeviceApp running on fxDevice should not be able normally to reboot the fxDevice). The required AL is generated by the fxSDK and included in the manifest files. It may also be published in the fxStore for that give application. A preset of ALs could be defined to categorize the applications security risks.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0147

Software Security Verification

[0096] It is important for the target network element to verify the authenticity and integrity loaded software prior to install. The uniquely defined security keys of the application developers sign the applications. The following rules may be followed to ensure software securityverification:

Only the verified OS/Firmware software can install on verified hardware;

Only the verified OS/Firmware software can run on verified hardware (continuous verification of essential parts of the software);

Only the verified fxDeviceApp software can install on verified hardware;

Only the verified fxDeviceApp software can run on verified hardware (continuous verification of essential parts of the software).

[0097] Each fxDeviceApp is accompanied by a manifest that is generated by the Integrated Device Electronics (IDE) at the time of compilation of the fxDeviceApp. The manifest file specifies which libraries and frameworks the fxDeviceApp uses. The fx:Manager uses this information to determine the security risk of the fxDeviceApp as well as the required Access Level (AL) to run this fxDeviceApp. In order to ensure that manifest file is actually genuine manifestation of the fxDeviceApp and its integrity remains intact, a hashing algorithm could be used to generate a signature. An example is shown below:

Signature= HMAC-SHA256(K, Message);

where

Message= fxDeviceApp + Manifest

K = shared secret key between fxStore and fxManager.

[0098] The signature is generated by the IDE/fxSDK at the compile/build phase and is packaged with application binary and the manifest file. Once the distributed application package (dAP) is downloaded into the fxManager (Application Repository), the components of the signature, application binary, and manifest file are unpackaged. The fxManager then uses the same hashing algorithm to calculate

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0148

the signature as above using the pre-shared key (K). If the calculated signature and the unpackaged signature match, it would prove the authenticity of the manifest file and the integrity of both manifest file and the fxDeviceApp.

[0099] Code Generation Process (Compilation):

fxDeviceApp compiled Manifest file generation Signature calculation Create fxDeviceApp Package

[0100] Install Process(Consumption):

Un-package fxDeviceApp Signature calculation Signature comparison [0101] This model works if the key K can be shared with two trusted parties (fxManager and fxStore). The manifest file that is generated by the fxSDK shall be secured by integrity checks to ensure it remains intact throughout the transaction (i.e. from compilation to consumption).

Communication Security

[0102] Security mechanisms built into the fxDeviceApp-0 and fxCloudApp-0 forces the fxDevice to only use the configured fxCloud servers for Software Security Verification, fxDeviceApp downloads, and fxDeviceApp-fxCloud communication. Optionally, the fxDeviceApp-fxCloud communication can be customized per application requirements. One implementation of this interface could use SPDY Protocol or per-application VPN.

[0103] For the purpose of inter-App communication, an Inter-App Communication (IAC) policy template is applied to an fxDeviceApp that defied the communication policies between the fxDeviceApps and fxCloudApps. The network administrator may use one of the pre-configured IAC policy templates (exemplified below):

Isolated: this type of applications cannot communicate with any other application in the same fxDevice and does not have any associated fxCloudApp;

Isolated-Extended: this type offxDeviceApps cannot communicate with any other fxDeviceApps in the same fxDevice but could communicate with its associated fxCloudApp;

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0149

Private: a private fxDeviceApp in one fxDevice could communicate with same private fxDeviceApps in other fxDevice but does not have any associated fxCloudApp;

Private-Extended: a Private fxDeviceApps in one fxDevice could communicate with same private fxDeviceApp in other fxDevice and has its associated fxCloudApp;

Community: a group offxDeviceApps that belong to the same community can communicate with each other;

Promiscuous: a Promiscuous app could communicate with any other Promiscuous application and Community application;

Custom: the network administrator could create custom IAC Policy templates and could include (as examples):

Template name;

Inherited pre-configured template if any;

The protocols allowed to use for communication;

The maximum data rate allowed for inter-app communication between a pair of fxDeviceApps or between an fxDeviceApps and an fxCloudApp;

Policies related to the other fxDeviceApps in the same fxDevice, in other fxDevice, associated fxCloudApp and unassociated fxCloudApps;
Policies related to direct fxDeviceApp to fxDeviceApp communication or relayed via fxCloud communication.

Secure Boot

[0104] The secure boot refers to a capability where the main boot code is in the fxCloud with the following exemplary procedure:

The Boot Loader (BL) in the fxDeviceApp-0 of the fxDevice looks for a particular DNS (DNSSEC) name of the fxCloud. The network administrator should define the DNS entry in their network. If there is a private fxCloud, the resolved domain name points to the fxCloud. If using shared cloud, the resolved

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0150

domain name points to the cloud point of presence;

A secure tunnel (TLS) is created between the BL in fxDevice and the fxCloud using pre-burned X.509certificate;

BL identify itself to the fxCloud and include System Information;

fxCloud determines the appropriate boot file and points the BL to the exact address of the boot file in thefxCloud;

BL downloads the correct boot file;

BL runs Software Security Verification;

BL reboots the fxDevice and the system will be ready for operation.

Frameworks

[0105] The following describes an example of the API Framework for the dSDN. Some APIs are purely local to the fxDevice or the fxCloud and some may be extended from the fxDevice all the way to the fxCloud using dRS.

General Framework

[0106] This framework refers to general libraries and APIs inherited from the legacy OS. An example of the legacy OS may be Embedded Android®. This framework allows for general processing and may be used for algorithmic applications.

Wireless Framework

[0107] The Wireless Framework adds a set up functions related to managing and controlling the wireless module. Table 1 highlights some examples of methods/functions that may be available in the Wireless Framework.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0151

Table 1: Wireless Framework	
1v1emoa	uescnpt1on
discoverNeighbors	Discovers wireless neighbors and returns their identities
retrieveNeighborInfo	Retrieves detail information of neighboring BS/AP information. This method may transparently call cloud servers to gather more information
measureInterference	Measures wireless interference observed on a frequency or a set of frequency channels
setWirelessConfig	Sets various parameters in the wireless module such as power, frequency channels, Multiple Input/Multiple Output (MIMO) antenna configuration, etc.
definePHYReceiverAlgorithm	Defines the physical layer receiver algorithm
connectedDevice	Returns details about the connected mobile devices (length of connection, data tx/rx) to a given wireless interface (channel, band, sector)
forceOffload	Offloads connected mobile devices to other access technologies (e.g. from cellular to WiFi)
homeAreaNetworking	Control of home automation wireless technologies such as ZigBee or Z-Wave
setNeighborCells	Sets neighboring cellsinformation
forceHandoff	Handoffs connected mobile devices to the same access technology but different channel or frequency
topMovers	Identifies the mobile devices with highest degree of mobility

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0152

positionDevice	Accurate positioning of mobile devices which may include geo-fencing data
borrowSpectrum	Allows one BS to borrow spectrum from another
configureMACScheduler	Configures the MAC scheduler (for example prioritize a particular user)
loadBalance	Enables load balancing across radio channels
defineMACScheduler	Defines the MAC scheduler algorithm

Security Framework

[0108] The Security Framework allows the developers to use a specialized security functions/methods. Some of these new methods may use the legacy OS frameworks. Table 2 highlights some examples of methods/functions that may be available in the Security Framework.

Table 2: Security Framework

Method	Description
setDOSDetect	Detects DOS attacks on user plane as requested
configureFirewall	Provides firewall functions such as traffic filtering, permissions (this may use dRS)
defineDOSDetectAlgorithm	Allows replacing and redefining the DOS algorithm
encryptTraffic	Encrypts the content of a message or a class of traffic as requested
aumentIcateuser	Authenticate users asrequestea
integrityCheckContent	Verifies authenticity of a message using its checksum information
configuresIPS	Configures IPS functions on the user plane

FastPath Framework

[0109] The FastPath Framework allows the developers to identify and manipulate the data path with a DPI capability. Table 3 highlights some

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0153

examples of methods/functions that may be available in the FastPath framework.

Table 3: FastPath Framework	
Method	Description
matchTraffic	Identifies and matches the traffic type requested by the developer
aeiecn..,ongesuon	Detects if there is congestion on the data path
identifyTopURLs	Identifies the most visited URLs by the users
detectlocalConversation	Detect conversations and packet routing between the end user on the same fxDevice
detectDevice	Detects traffic of a given mobile user device
redirectTraffic	Redirect data traffic to a given network element. The traffic may be identified with a user or a class of traffic
connectedDevices	Returns all the connected mobile devices to a given fxDevice
assIgnAddress	Allocates a particular IP address to the mobile device
setPriority	Set traffic priority to a class of traffic
markTraffic	Marks the user traffic with the desired QoS tags
setupTunnel	ets up data tunnel from fxDevice to a destination
Adscript	Adds a JavaScript to an active transiting web page
detectIdenticalFlows	Identifies the identical flows coming from the WAN/uplink interface on the fxDevice
aggregateFlows	Enables aggregation of flows to save WAN/uplink utilization
routeTunnel	Routes a packet or a class of traffic onto an established tunnel
identifyTopDevices	Identifies users that generate most traffic load

Messaging Framework

[011O] The Messaging Framework allows the developers to send messaging between the applications residing in the fxDevice and fxCloud. Table 4 highlights some examples of methods/functions that may be available in the Messaging Framework.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0154

Table 4: Messaging Framework	
Method	Description
sendIntraApp	Sends an Intra-App message from one fxDevice to another
sendInterApp	Sends an InterApp message from one application to another
sendfxCloud	Sends a message to the fxCloud
sendSMS	Sends an SMS to the user of choice

Caching Framework

[0111] The Caching Framework allows the developers to cache particular contents or files that are accessed frequently in the caching engine. The caching can be done locally in the fxDevice, clustered cache (shared amongst a few fxDevice), or cloud caching. Table 5 highlights some examples of methods/functions that may be available in the Caching Framework.

Table 5: Caching Framework

Method	**Description**
storeForward	**Enables Store and forward model for a target class of traffic where the target traffic is matched and stored upon a condition and forwarded the condition is relieved**
cacheWebSite	**Caches the most visited web sites**
cacheVideo	**Caches the most viewed videos**
cacheCloud	**Caches the requested content or traffic type in the cloud storage in fxCloud**

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0155

Management Framework

[0112] The Management Framework enables the developers to manage the BS platforms and perform administrative procedures. Table 6 highlights some examples of methods/functions that may be available in the Management Framework.

Table 6: Management Framework

Method	Description
upgradeFirmware	Started and executes the firmware upgrade procedure on the fxDevice
upgradeOS	Started and executes the operating system upgrade procedure on the fxDevice
remoteBootup	Performs remote boot up process where the fxDevice works closely with the fxCloud to perform the bootup
remotePower	Allows fxCloudApps to remotely turn on and off a fxDevice or elements in the fxDevice (e.g. the wireless)
gatherStatistics	Instructs fxDevice to gather specific statistics. Various commands may be consolidated for gathering statistics. Number of active users Stats of time spent by users in the cell Stats of quality perceived by the users Neighboring cell info Overall traffic passed through the system Total number of connected devices per frequency, per carrier, per site, per location Handover failure statistics Cloud aggregated data processing and full report Total traffic passed Applications used (meta-data correlated) Bandwidth consumed User's mobility pattern

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0156

setIvIorntonng	Allows the developer to get reports and statistics of the fxDevice traffics, errors, and neighboring environment. The reporting criteria could be set up as: 1) Period, 2) Event based (reports only if a threshold is passed; a threshold could be based on absolute numbers or deltas), 3) One-time (i.e. the information is pulled once). The exact parameters to get report on may use the same data structures used by the method "getStatistics".

getSystemInfo	Retrieves static and dynamic system info such as: Hardware info: vendor, model number, capability profile. GPS, Cellular, WAN interfaces, RFID/NFC Heat temperature • Vendor Specific Information • Software info: OS version, firmware information • CPU load, per-application load (CPU, memory, accelerators) Loaded apps and status Memory utilization • Storage utilization Interface status & utilization (RF, backhaul, management...)
getfxDeviceLocation	Retrieves location of the fxDevice Location would be presented in GPS coordinates and civil address Location enhancement: Cloud, map data, and neighbor discovery could help more accurate positioning
getPowerConsumption Rate	Retrieves current power consumption rate of the fxDevice
configureWAN	Configures the WAN settings
retrieveNeighborInfo	Retrieves neighbor information from the fxCloud
getRFIDInfo	Returns the RFID/NFC information of the fxDevice
setRFIDInfo	Sets the RFID/NFC information of the fxDevice

Extensible API Framework

[0113] This framework allows the developer to create custom APIs and make it accessible for other applications (in the fxDevice or in the fxCloud). In turn, the fxCloud could present these APIs using e.g. REST technologies (via the fxCloud

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0157

Northbound Interface) to the developer outside the dSND system. There might be limits put on the APIs exposed through the cloud to avoid potential misuse or security threats. The dRS enforces the security policies defined by the admin (via fxManager). The applications take a role of Client Application (CA) or Server Application Role (SA). The CA and SA could be distributed in the fxDevice or fxCloud. The CA makes requests and SA serves the requests.

[0114] The Extensible APIs could be categorized (as examples) into:

Read-Only Data: this type of APIs allows a CA to read data from SAs without the ability to change or request any particular action from the SAs

Read-Write Data: this type of APIs allows CA application to read data from SAs with the ability to change some of their data but no option for requesting any particular action fromSAs;

Procedural: this type of APIs allows CAs to execute a particular procedure in SAs. The rights to change any data still depends on whether the API allows Read-Only Data or Read-Write Data types.

Cloud Management Framework

[0115] The Cloud Management Framework presents a collection of methods enables management of the cloud services and resource. Table 7 highlights some examples of methods/functions that may be available in the Cloud Management Framework.

Table 7: Cloud Management Framework

Method	Description
detectCloudCongestion	Detects congestions on a particular VM
deleteVM	Deletes an active VM permanently
suspen	Suspends an active VM but doesn remove
activate	Activates an already created VM
createVM	Creates a new VM but doesn't activate it

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0158

EXAMPLES OF DAPPS

[0116] The dSDN create an end-to-end programming platform and the possibilities of vApps are only limited by the developers' imagination. An example list of potential vApps are presented in Table 8 below.

Table 8 highlights examples of use cases possible by the Distributed Software Defined Network (dSDN)

use	Description	Framework Involved/used
1	Distributed SON: For example, the WiFi AP could search for the least crowded channel to minimize the interference.	Wireless {discoverNeighbor, retrieveNeighborInfo, measure nterference, setWirelessConfig, loadBalance}, General Framework, Messaging {sendIntraApp}
2	BS Hot Firmware Upgrade without impact to the basic functions of the BS.	Management (upgradeFirmware}
3	BS Hot infOS Upgrade without impact to the basic functions of the BS. For this function to support full hot upgrade without operational impact, the full platform should have dual processor.	Management (upgradeOS}
4	Secure network—based booting. This allows to keep the code securely in the cloud to avoid tampering by the ODM/CM or Man—in—the---Middle attacks.	Management {remoteBoot}
5	Turning on or off APs if nobody is in the office. Sensors in the building send information to fxCloudApp which in turn orders power control of the Aps.	Management {remotePower}
6	Store and forward model for low priority M2M applications where the target traffic is matched and stored in the case of network congestion and forwarded once the network congestion is relieved.	DataPath {matchTraffic, detectCongestion}, Cache {storeForward}
7	Replacing MAC Scheduler	Wireless {defineMacScheduler}
8	Replacing Receiver Algorithms	Wireless {definePHYReceiverAlgorithm}
9	Caching most visited website or video clip in the enterprise	DataPath {identify TopURL}, Cache {cacheWebSite}
10	The most viewed videos and webcasts are cached in the wireless edge.	DataPath {identifyTopVideos}, Cache {cacheVideo}
11	Local Routing: data and voice sessions between two parties on the same cell are routed within the same cell bypassing the core Network.	DataPath {detectLocalConversation, RouteLocal}
12	Local Routing: data and voice sessions between two parties on the nearby cells are routed locally bypassing the core network.	DataPath {detectConversationDetect, routeLocal}
13	Temporary redirection of a data flow for monitoring and security reasons. In this use case, a particular use(s traffic is tracked by redirection to a monitoring station. If user is mobile, such profile of tracking and associated redirection is transferred from fxDevice to fxDevice either directly or via the fxCloud.	DataPath {detectDevice, matchTraffic, redirectTraffic}
14	Power Calendar: On—demand coverage (triggered by passing traffic thresholds in certain cells or by calendaring): Operators need to design their network for peak usage rates at high costs, leaving their network underutilized most of the time.	DataPath {detectCongestion}, Management {remotePower}
15	DOS attack recognition and action could be blocking the user, the app, and/or notifying the user via an SMS message.	Security {setDOSDetect, configureFirewall}, Messaging {sendfxCloud}
16	Applying certain ACLs to the BS. For example, the students in the class can only access Facebook between the breaks and access to Facebook is blocked during the class.	Security {configureFirewall}
17	Forced offloading of some cellular users (users that have attached to the same cell for a while) to WiFi by changing "something" in the cellular connection to force WiFi connection (Heteroaeneous)	Wireless {identifyConnectedDevice, forceOffload}
18	Service continuity and anchoring at eNB for WiFi using the same LTE IP address (Heterogeneous).	DataPath {identifyConnectedDevices, assignAddress}
19	Graphic/general processing for special use cases such as City BS	General Frameworks
20	Application filtering/throttling (secondary conditions: time—based, location—based, subscriber—type, none).	DataPath {matchTraffic}, Security {configureFirewall}
21	Local Services: The enterprise user can discover Bonjour or UPnP devices and services (e.g. printer) even when connected to LTE/3G small—cell in the enterprise. In this case, the small cell or fxDevice pro—actively listens to Bonjour and UPnP discovery and advertisement messages (SSDP) and cache the available services until a device request for such information. In the case of Bonjour, fxDevice could listen to Multicast DNS (mDNS) messages sent over cellular connection (LTE/3G) and multicast them on the LAN or WLAN interfaces.	DataPath {matchTraffic, routeLocal}
22	Local Services: The premise—owner (e.g. airport, hotel} can provide ads under the browser for sponsored WiFi or cellular (3G/LTE) access over multi---mode small cell. The premise owner could use this service to advertise special offers or third party ads. This technique uses HTML <script> for example to add special content to the bottom of the page.	DataPath {matchTraffic, addScript)

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0159

23	Distributed Content Distribution Network (CON). In this use case, the fxDevices function as CON that bring the contact closer to the end **users.**	DataPath {matchTraffic}, Cache {cacheVideo}
24	**In case of national security, the service provider may limit the access** to the network so only the law enforcement can use the cellular network. In this case, the admin defines a Target Area (TA) using an fxCloudApp in the fxCloud where the emergency lockdown needs to be applied. The IMSI/MSISDN number of law enforcement mobile devices are sent to the fxDevice in the TA which in turn enforce the policy. The policy may allow the public to just sent SMS while the law enforcement could have full access and priority to the system.	DataPath {matchTraffic, setPriority}, Security {configureFirewall}
25	Emails and web browsing monitored by the cloud services assisted by the agent in the dSDN network. In this case, the fxDevice detects certain traffic (email or web in this case) and redirects them to a **cloud service for security cleansing**.	DataPath {matchTraffic,redirectTraffic}
26	Analytics pre----analysis: The data collected from M2M devices could be processed (noise could be filtered out) and packaged (compressed) for consumption by cloud services.	DataPath {matchTraffic}, General Framework, Messaging {sendfxCloud}
27	Certain traffic could be marked with QoS DSCP codes for further processing in the network.	DataPath {matchTraffic,markTraffic}
28	Proprietary SP services and applications that run purely between the phone and the BS . For example, the SP may implement a certain application for the law enforcement to get direct feed of surveillance cameras in a target area (TA).	DataPath {matchTraffic}, Security {configureFirewall}, General Framework
29	IP—PBX implementation in the small---cell or WiFi AP for enterprise use. This would allow a simple out of a box solution. The VoIP phones connect to the fxDevice via the LAN connections. The WAN connection could connect to an fxCloudApp for PSTN calls.	DataPath {matchTraffic}, GeneralFramework
30	Network management system can add its own agent/probe to collect and consolidate data and transmit it to its cloud element using any protocol.	Management {gatherStatistics, getSystemInfo, control}, GeneralFramework
31	Application based performance monitoring on the wireless link (number of packets sent in idle mode, retransmit rate...). Such information could be shared with the app developer to optimize its application.	Management {gatherStatistics}, GeneralFramework
32	Seamless network sharing and mergers. This is the case where two Network Operators would like to share or merge their network. Using dSDN's infManager, the network administrators could merge the network administration control. This process may include setting up the desirable default fxApp packages of the network operator.	fxManager Portal Service
33	**Following a user to understand where it faces coverage issues or** call drops. Once discovered, the system could send a message to the user acknowledging an issue and that the SP is trying to fix this. The information is uploaded to the cloud for further analysis.	Management {gatherStatistics}, General Framework, Messaging {sendSMS}
34	Reporting: This allows the developer to get reports and statistics of the fxDevice traffics, errors, and neighboring environment. The reporting criteria could be set up as: 1) Period, 2) Event based r i'c, on i :r!h<i)£a J rE(is:ot' /i ,,,;'!ti)ed on	Management {setMonitoring}, GeneralFramework
35	Proprietary security protocol between the application in the mobile device and the BS in enterprise deployment. This would allow SP to create enterprise----class LTE for indoor using small cells.	Security {encryptContent, authenticateUser, Integrity, configureFirewall}, DataPath {RouteIocal}, General **Framework**
36	The femto cell at home that can't scan neighboring cells, gets a list of neighboring cells from the fxCloud or neighbors to broadcast to improve outbound handoffs.	Wireless {discoverNeighbors, retrieveNeighborInfo, setNeighborCells}, Management {retrieveNeighborInfo}
37	IMS---capable femto enabling various SP services to initiate or transferred between home network and mobile devices.	Messaging {sendfxCloud}, General Framework
38	Integrated smart home with multi---mode femto/WiFi router. For example, the application in the smart home router (fxDevice) allows for coordination of home physical security service (such as ADT services). With the fxCloudApp, the user could control the security status of his/her home and remotely control the setting.	Wireless {homeAreaNetworking}, Messaging {fxCloud}, General Framework
39	Home CCTV DVR implementation in an integrated home router. The **camera feeds are recorded on the same router caching engine.**	Cache {cacheVideo}, Messaging {sendfxCloud}, General Framework
40	Enterprise CCTV DVR implementation in an integrated router with cloud backup. The camera feeds are recorded in the cloud.	Cache {CacheVideo, CacheCloud}, Messaging {fxCloud}, General Framework
41	User traffic separation into various MPLS/LSP (integrating VPN and BS). In this case, users are classified into groups and each group is tunneled to the core network. For example, law enforcement use the encrypted tunnel while the other users use the defaulttunnel.	DataPath {Match, SetupTunnel, RouteTunnel}
42	**Dynamic spectrum allocation where a BS borrows unused spectrum** from the neighboring BS in case of temporary congestions.	DataPath {detectCongestion}, Wireless {discoverNeighbors, setWirelessConfig}, Management {retrieveNeighborInfo, borrowCarrier}, Messaging {fxCloud}

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0160

43	Proxy ANDSF (Access Network Discovery and Control Function) in the BS. When the UE tries to access the ANDSF, the Proxy ANDSF fxDeviceApp in the BS intercept the ANDSF messages and replies on behalf of the ANDSF. This would speed up the WiFi network discovery for example.	DataPath {matchTraffic}, General Framework, Messaging {sendfxCloud}
44	HD Audio calling: audio codecs in BS allows the phones with the right capabilities to communicate with HD Audio quality calling. If one of the mobile devices can't handle the HD quality audio, the BS performs the transcoding of the audio.	DataPath {matchTraffic}, General Framework
45	Enabling mobility on White Space Spectrum: The latest white space database will be downloaded. The mobile devices with special software applications make requests to the BS to retrieve the local available white space frequencies. This would allow the mobile device to handoff seamless between BS using the white space spectrum. The BS would advertise such a capability to the UEs. The **secure communication allows peer—to—•Peer communication** between the UE and the BS for certain applications such as White Space Channel lookup or site acquisition tools for field engineering and RF planning.	Messaging {sendfxCloud}, General Framework
46	**Site acquisition tool where the technician uses a software tool on his** mobile phone to interact with BSs to determine the best place to deploy the new BS. The phone app interests with fxDeviceApps in fesrf! d : . r J g i v e,,x J'iiiy : ;ar\?.;' Rtr:[';fms air	Messaging {sendfxCloud}, General Framework. DataPath {matchTraffic, setPriority}
47	DNS caching at BS to accelerate DNS lookup for subsequent users. **This would improve the user experience by reducing response** delays.	DataPath {matchTraffic}, General Framework
48	Continuous SLA monitory for certain protocols, applications or general link quality for jitter, delays, bandwidth.	DataPath {matchTraffic}, Management {gatherStatistics}, General Framework
49	Enterprise users can use the small cell deployed in their enterprise free of charge. The DPI engine in fxDevice classifies the traffic and identifies the enterprise private data and they would be routed locally and charging records are generated by the BS (the SP decides how to bill the enterprise user and it may decide to make such usage.	DataPath {matchTraffic, routelocal}
50	On—premise small—cell aggregation where on fxDevice takes the responsibility of aggregating multiple small—cells on the premise and presenting them as a single BS/Node—B/eNodeB to the core network. In this case the aggregating fxDevice (AfxE) may take responsibility of local radio resource manager (RRM) or self—optimizing network (SON) server. AfxE can also implement and **enforce policies concerning local service access on the premise** (enterprise) as well as enable seamless handover between cellular technologies and on—premise (enterprise) local area network (LAN) for example using WiFi technology.	General Frameworks, Wireless {discoverNeighbors, retrieveNeighborInfo, setNeighborCells, setWirelessConfig}, Messaging {sendIntraApp}
51	Self adjusting fxCloud compute resources: The fxCloud creates and tears down virtual machines (VMs) based on the traffic load measured by itself or the packet processing and data plane engine. It is also possible that the VM adjustments are made based on the rush hour or pre—configured hours of the day or manually by the **administrator**.	CloudManagement {removeVM, createVM, detectCloudCongestion, actiavteVM, suspendVM}
52	Core Network Function Virtualization (cNFV) using distributed Software Defined Networking (dSDN): In the case, once an fxDevice is commissioned, its fxDeviceApp0 (i.e. firmware) uses predefined DNS names to discover the rest of the core network (e.g. MME, S—GW, etc.). The core network elements could be created.	General Frameworks, Messaging {sendInterApp, sendIntraApp), CloudManagement {deleteVM, suspendVM, createVM, activateVM, detectCloudCongestion}
53	Smart display and digital signage as wireless base stations: this use case allows combining the capabilities of a base station and the smart display in shopping malls, airports, and enterprises. There is a already electric power and connectivity for the smart display that the **base station function could use**.	**General Frameworks**
54	Tracking a mobile user at base station level even if the UE is in the idle mode. Currently in the idle mode, the MME, SGSN, or VLR/MSC can only know the location of the UE in with a accuracy of LA/RA/URA which is a very large area. The administrator uses the fxCloudApp and requests tracking of the user/UE (the admin may provide target area where the user/UE may be located; e.g. zip code or town name). The fxCloudApp contacts its fxDeviceApp in the fxDevices in the target area (TA) requesting information of a particular IMSI/MSISD/TMSI. Once an fxDevice responds indicating knowledge of the user location, the fxCloudApp directs that serving fxDevice to inform the fxDeviceApp in the next fxDevice of the instructions to track the user and contact fxCloudApp when they user hands off to due to its mobility. fxCloud could create user friendly interfaces and presentations to show for example the direction the **user is moving, predict next location of the user, and estimate arrival of the user to a certain location**.	DataPath {matchTraffic}, Messaging {sendIntraApp, sendfxCloud}
55	In case of emergency, fxDevices in a target area block all public traffic and only allow law enforcement sessions using specially encryption and authentication between the UE and the fxDevice or fxCloud (depending on the requirements).	DataPath {matchTraffic, setPriority}, Security {configureFirewall, encryptTraffic, authenticateUser}, **General Framework**

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0161

56	SON triggered by the data traffic consumption. In this case, the mobile cell configuration is adjusted based on the current traffic of the cell. It could for example force handoff some users (high users or users moving fast) to neighboring cells.	DataPath {identifyTopDevices}, Wireless {forceHandoff}
57	Time based QoS and policy. In this use case, the infManager pushes certain policies in target area (TA). Such polices could be turning on extra capacity (frequency channels) or QoS policies and traffic prioritization on the WAN like.	DataPath {matchTraffic, setPriority}, Wireless {discoverNeighbors, setWirelessConfig}
58	Fast moving users detection: in this use case, the application in the BS measures the speed of the user (based on the time was handed off to this cell until it was handed over to the next cell). This information could be used by the fxDevice or fxCloud to force handoff the mobile user to a larger cell to reduce handoff rates.	General Frameworks, Wireless {identifyTopMovers, forceHandoff}
59	**Emergency broadcast of information and live camera feeds. In case** of emergency, the enterprise network administrator temporarily block all traffic and only allow emergency related traffic across the target area in the enterprise network. For example, the network admin broadcasts the camera feeds and alerts over the WiFiaccess points using multicast IP techniques. The client devices and digital sinages can listen to that particular multicast and show the camera feeds and emergency alerts.	DataPath {matchTraffic, setPriority}, Security {configureFirewall}, General Framework
60	Dynamic advertising in shopping malls. The shopper would use an app from the shopping mall. When the shopper brings up or refreshes the app, the dSDN allows the shopping mall to pinpoint the exact indoor/outdoor location of the user (e.g. the shop the shopper is in) and send special coupon and ads for that store to the shopper.	General Frameworks, Wireless (positionDevice}, Messaging {sendIntraApp, sendfxCloud}, Security {configureFirewall}, DataPath {matchTraffic, redirectTraffic}
61	Smart tour guide. The tourist would use an app from the tourism board or museum. Wihen the user brings up or refreshes the app, the dSDN allows to pinpoint the exact location of the tourist (e.g. the painting he/she is looking at) and send tour guide information (e.g. audio, video, hypertext). dSDN allows to localize the positioning.	General Frameworks, Wireless (positionDevice}, Messaging {sendIntraApp, sendfxCloud}, Security {configureFirewall}, DataPath {matchTraffic, redirectTraffic}
62	General hotspot landing page	DataPath {matchTraffic, redirect Traffic}, Security {authenticateUser}
63	Proprietary wireless mesh implementation	Wireless {discoverNeighbor, retrieveNeighborInfo, measureInterference, setWirelessConfig}, General Framework, Messaging {sendIntraApp}
64	Power and energy consumption map of the network. A real-—time fxCloudApp pulls the power consumption details from each fxDevice periodicaly or on demand. The pulled data then are combined with the location information and correlated into a map to present a **viewable consumption map**.	Management {getfxDeviceLocation, getPowerConsumptionRate}
65	Network deployment optimization. This application allows network operator to get advice for best optimal reconfiguration and/or redeployment of the network. For example, the network admin could get suggestion that based on the power consumptions and coverage perceived and spectrum utilization, it might be better to replace a particular group of smaller cells with a single macro-—-cell. This app could also pull information from larger OEMs based on the subject deployment scenario (the OEMs either provide this information statically to the tool or provide network APIs for use by this app e.g. using REST protocol).	Management {getfxDeviceLocation, getPowerConsumptionRate, gatherStatistics}
66	High availability. This app allows two or more fxDevice to form a redundancy cluster where they participate in statueful redundancy and load-—-balancing.	Messaging {sendIntraApp}, General Framework
67	Bandwidth Bursting. This feature enables the dynamic adjustment of backhaul or WAN link bandwidth as the demands change in the enterprise or public wireless deployments. In this case, the fxDeviceApp in the fxDevice monitors the WAN link utilization, once a preset threshold percentage has reached, it messages its sister fxCloudApp which in turn would send a request to the fxDeviceApp to increase or decrease the WAN bandwidth. If additional backend reconfiguration is needed, the fxCloudApp performs those additional changes prior to informing the fxDeviceApp of the bandwidth.	Management {getStatistics,configureWAN}
68	Wireless link quality measurement by the small-—cell or WiFi AP: The wireless Soc in the small-—-cell or WiFi AP continuously monitors the link quality for power adjustment and handover reasons. Such information could be exposed to the app developers to create cell-—-level analytics and reports without the need to have test-— drives or field testing of service quality.	Management {gatherStatistics}, GeneralFramework
69	Data Path Consolidation: An application on the fxDevice discovers **identical data feeds from a cloud server to two or more users. The** fxDeviceApp in the fxDevice can discover this potential duplicate flows at the session establishment (via a proxy function) and it would consolidate those sessions into one on the uplink towards the cloud server. A good example of this use case is the web—based desktop sharing where different many users in the same office site get the same feed from the cloud server.	FastPath {detectIdenticalFlows, aggregateFlows}

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0162

[0117] The following references are herein specifically incorporated by reference in their entirety into this disclosure:

1 3GPP TS 23.002, "NetworkArchitecture";

2 3GPP TS 23.402, "Architecture enhancements for non-3GPP accesses";

33GPP TS 23.401, "Genera! Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access";

4 Open Base Station Architecture Initiative (OBSAI); and

[5] SPDY Protocol (b-Hp:_Ig_c1tc1_tn_cxtLi_E=1L999_Qqg r c1ftn1>_§_O_E=i:_Utp_1>i§_§>_dy().

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0163

CLAIMS

1. A system for processing data packets in a network, the system comprising:

a programmable network device hosting a plurality of first network applications;

a programmable cloud device hosting a plurality of second network applications;

wherein at least one of the plurality of first network applications in the programmable network device and at least one of the plurality of second network applications in the programmable cloud device are in secure communication with each other to form a distributed application;

the programmable network device and programmable cloud device are each powered by a sandboxing operating system which facilitates deployment of the plurality of first and second network applications and facilitates upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device;

wherein the programmable network device verifies authenticity of the upgrades to the plurality of first network applications and the programmable cloud device verifies authenticity of the upgrades to the plurality of second network applications and wherein the verification is based on unique security keys associated with each of the plurality of first and second network applications;

an application repository storing distributed applications which have been tested for installation in the programmable network device and programmable cloud device; and

an application management portal coupled to the programmable network device, programmable cloud device and the application repository, wherein the application management portal manages usage of the distributed applications on the programmable network device and programmable cloud device.

2.The system of claim 1, wherein each of the programmable network device and programmable cloud device include an access network interface unit to send and receive communications and a processor with a memory associated with the network interface unit.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0164

3.The system of claim 1, wherein the at least one of the first network applications is responsible for a plurality of functional capabilities of the programmable network device and is capable of at least one of the group consisting of: updating the firmware of the programmable network device; updating default communication protocols of the programmable network device; initializing programmable network device settings; boot loading the programmable network device; providing security digital certificates; providing operating system **(OS)** and kernel upgrades; and upgradin-g and commissioning agents for upgrades of the at least one of the first network applications.

4.The system of claim 1, wherein at least one of the second network applications is responsible for a plurality of functional capabilities of the programmable cloud device and is capable of at least one of the group consisting of: upgrading and commissioning agents for upgrades of the at least one of the second network applications and related procedures; hypervisor upgrades; virtual machine and isolated execution environment lifecycle management; security digital certificate management; operating system **(OS)** and kernel upgrades; and default communication protocols.

5.The system of claim 1, wherein the programmable network device is a Long Term Evolution (LTE) evolved NodeB (eNB) and the programmable cloud device hosts LTE core network functions in separate virtual machines, the network functions consisting of: Mobility Management Entity (MME), Serving Gateway (S-GW), and Packet Gateway f'-GW).

6.The system of claim 1, wherein the programmable network device and programmable cloud device further comprise:

a virtual fabric which provides a secure communication layer for at the least one of the plurality of first network applications and the at least one of the plurality of second network applications.

7. The system of claim 1, further comprising:

a second programmable network device which includes the at least one of the

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0165

plurality of first network applications which is in the first programmable network device.

8.The system of claim 1, wherein the programmable network device and programmable cloud device further comprise:

a distributed resource service (DRS) which controls access to a plurality of resources by the plurality of first network applications and the plurality of the second network applications upon receiving instructions from the application management portal.

9.The system of claim 1, wherein the plurality of resources include at least one of a group consisting of: a firewall; data and statistics; storage; compute; load evaluator; general settings; routing engine; virtual machine master controller; power controller; and wireless engine.

10.The system of claim 1, wherein the programmable network device and programmable cloud device further comprise:

a distributed resource service (DRS) which is capable of providing at least one service from a group consisting of: exposing application programming interfaces (APIs) to other applications; configuring and managing platform resources; policy enforcement and authorization of applications access to platform resource and other application's APIs; and policy conflict resolution.

11.The system of claim 1, wherein the programmable network device and programmable cloud device further comprise:

a distributed content service wherein contents generated by the at least one of the plurality of first network applications may be shared with the at least one of a plurality of second network applications.

12.The system of claim 1, wherein the programmable network device and programmable cloud device further comprise:

a distributed notification service wherein notice of a particular event is sent from at least one of the plurality of first network applications to at least one of the plurality of

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0166

second network applications.

13.The system of claim 10, wherein the DRS further includes a load controller; and wherein the load controller is adapted to monitor loads on at least one of the plurality of first network applications and at least one of the plurality of second network applications and effect change in accordance with thresholds received from the application management portal.

14.The system of claim 1, wherein the programmable network device, at least one of the plurality of first network applications, the programmable cloud device, and at least one of the plurality of second network applications each have unique identities.

15. A system for processing data packets, the system comprising:

a programmable network device processing data flows hosting at least one of a plurality of first network applications;

a programmable cloud device having a plurality of virtual machines wherein at least one of the virtual machines hosts at least one of a plurality of second network applications;

wherein the at least one of the plurality of first and the at least one of the plurality of second network applications are in secure communication to form a distributed application;

the programmable network device and programmable cloud device are each powered by a sandboxing operation system which facilitates deployment of the plurality of first and second network applications and facilitates upgrade of the plurality of first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device;

wherein the programmable network device verifies authenticity of the upgrades to the plurality of first network applications and the programmable cloud device verifies the authenticity of the upgrades to the plurality of second network applications and wherein the verification is based on unique security keys associated with each of the plurality of first and second network applications;

an application repository storing distributed applications which have been tested

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0167

for installation in the programmable network device and programmable cloud device; and

an application management portal coupled to the programmable network device, programmable cloud device and the application repository, wherein the application management portal manages provisioning, usage and de-provisioning of the distributed applications on the programmable network device and programmable cloud device.

16.The system of claim 15, wherein the programmable network device is a Long Term Evolution (LTE) evolved NodeB (eNB) and the programmable cloud device hosts LTE core network functions in separate virtual machines, the network functions consisting of: Mobility Management Entity (MME), Serving Gateway (S-GW), and Packet Gateway f'-GW).

17. A system for processing data packets in a network, the system comprising:-

a programmable network device processing data flows, hosting at least one of a plurality of first network applications;

a programmable cloud device partitioned into a plurality of zones, wherein at least one of the plurality of zones hosts at least one of a plurality of fi.fst second network applications;

wherein the at least one of the plurality of first network applications and at least one of the plurality of second network applications are in secure communication to form a distributed application;

the programmable network device and programmable cloud device are each powered by a sandboxing operation system which facilitates deployment of the plurality of first and second network applications and facilitates upgrade of the plurality of first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device;

wherein the programmable network device verifies authenticity and integrity of upgrades to the plurality of first network applications and the programmable cloud device verifies the authenticity of the upgrades to the plurality of second network applications and wherein the verification is based on unique security keys associated with each of the plurality of first and second network applications;

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0168

an application repository storing distributed applications which have been tested for installation in the programmable network device and programmable cloud device; and

an application management portal coupled to the programmable network device, programmable cloud device and the application repository, wherein the application management portal manages usage of the distributed applications on the programmable network device and programmable cloud device.

18. The system of claim 17, wherein the programmable network device is a Long Term Evolution (LTE) evolved NodeB (eNB) and the programmable cloud device hosts LTE core network functions in separate virtual machines, the network functions consisting of: Mobility Management Entity (MME), Serving Gateway (S-GW), and Packet Gateway f'-GW).

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0169

ABSTRACT

The Distributed Software Defined Network (dSDN) disclosed herein is an end to-end architecture that enables secure and flexible programmability across a network with full lifecycle management of services and infrastructure applications (fxDeviceApp). The dSDN also harmonizes application deployment across the network independent of the hardware vendor. As a result, the dSDN simplifies the network deployment lifecycle from concept to design to implementation to decommissioning.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0170

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
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3

CONFIRMATION NO. 2943

112449
Roark IP
1438 Dahlia Loop
San Jose, CA 95126

FORMALITIES LETTER

OC000000123462715

Date Mailed: 02/18/2021

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

An application number and filing date have been accorded to this application. The application is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given **TWO MONTHS** from the date of this Notice within which to correct the informalities indicated below. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

The required item(s) identified below must be timely submitted to avoid abandonment:

- A substitute specification in compliance with 37 CFR 1.52, 1.121(b)(3), and 1.125, is required. The substitute specification must be submitted with markings and be accompanied by a clean version (without markings) as set forth in 37 CFR 1.125(c) and a statement that the substitute specification contains no new matter (see 37 CFR 1.125(b)). The specification, claims, and/or abstract page(s) submitted is not acceptable and cannot be scanned or properly stored because:
  - The application papers (including any electronically submitted papers) are not in compliance with 37 CFR 1.52 because pages 54 contain text that is unreadable or of insufficient clarity.
    Application papers (including any electronically submitted papers) must be presented in a form having sufficient clarity and contrast between the background of the paper and the writing thereon to permit the Office to electronically reproduce the papers by use of digital imaging and optical character recognition. See 37 CFR 1.52(a)(1)(v).

Applicant is cautioned that correction of the above items may cause the specification and drawings page count to exceed 100 pages. If the specification and drawings exceed 100 pages, applicant will need to submit the required application size fee.

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice". https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at http://www.uspto.gov/ebc.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/bzewdie/

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0172

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
17/142,983	01/06/2021	2425	830	TAAG001CONT3	18	3

CONFIRMATION NO. 2943

112449
Roark IP
1438 Dahlia Loop
San Jose, CA 95126

FILING RECEIPT

OC000000123462714

Date Mailed: 02/18/2021

Receipt is acknowledged of this non-provisional utility 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 FIRST INVENTOR, 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 corrected Filing Receipt, including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction 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 provided that the request is grantable.

Inventor(s)

Pouya Taaghol, San Jose, CA;
Vivek Ramanna, San Jose, CA;

Applicant(s)

Big Data Federation, Inc., Santa Clara, CA;

Power of Attorney: The patent practitioners associated with Customer Number 112449

Domestic Priority data as claimed by applicant

This application is a CON of 16/900,963 06/14/2020 PAT 10893095
which is a CON of 15/836,824 12/09/2017 PAT 10686871
which is a CON of 14/295,331 06/04/2014 PAT 9843624
which claims benefit of 61/834,807 06/13/2013

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: No

Permission to Access Search Results: No

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 02/17/2021

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 17/142,983**

Projected Publication Date: Request for Non-Publication Acknowledged

Non-Publication Request: Yes

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

Distributed Software Defined Networking

Preliminary Class

725

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

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 (1-866-999-4258).

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 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 U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop

technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0176

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 17/142,983

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	80		N/A	
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	350		N/A	
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	400		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	18 minus 20 =	*	x 50 =	0.00	OR		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3 minus 3 =	*	x 240 =	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	830		TOTAL	

APPLICATION AS AMENDED - PART II

		(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($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	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	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
AMENDMENT B	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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Taaghol
Serial No. 17/142,983
Filed: 1/6/2021
Attorney Docket No.: TAAG001CONT3

Examiner:
Art Unit:

For: DISTRIBUED SOFTWARE DEFINED NETWORKING

Mail Stop Missing Parts
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

RESPONSE TO NOTICE TO FILE CORRECTED APPLICATION PAPERS

Applicants offer the following response to the Notice to File Corrected Application Papers mailed on February 18, 2021. Applicant has deleted the grey text on page 54 and inserted the proper color. A clean copy and marked up copy of the specification have been submitted. No new matter has been added.

Respectfully submitted,

ROARK IP

By: /Christopher J. Horgan/

Christopher J. Horgan
Registration No. 40,394
1438 Dahlia Loop
San Jose, CA 95126
Telephone: (408) 687-8306

Date: April 19, 2021
Attorney Docket: TAAG001CONT3

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0178

DISTRIBUTED SOFTWARE DEFINED NETWORKING

CROSS-REFERENCE TO RELATED PATENT APPLICATIONS

[0001] The present invention claims priority from U.S. Provisional Patent Application Serial No. 61/834,807, filed June 13, 2013 the disclosure of which is herein specifically incorporated by reference in its entirety.

FIELD OF THE INVENTION

[0002] This invention relates to a network architecture that facilitates secure and flexible programmability between a user device and across a network with full lifecycle management of services and infrastructure applications.

BACKGROUND

[0003] Existing network solutions are built on proprietary hardware and software. Network Operators & Information Technology (NOIT) can only configure the proprietary network infrastructure provided by vendors and are unable to add new and customized features or capabilities. As a result, the NOIT can only add new desired features by making such requests to their infrastructure vendors or pursuing standardization processes. But these existing approaches are time consuming and resource intensive.

SUMMARY OF THE INVENTION

[0004] An aspect of the disclosure herein is a plurality of network elements in a packet system, said network elements comprising: a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp; a flexible cloud platform having at least one fxCloudApp, wherein the at least one fxDeviceApp and at least one fxCloudApp are in secure communication to form a distributed application; and an application management portal which manages life cycle of the distributed applications.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0179

[0005] Another aspect of the disclosure is a plurality of network elements in a packet system, said network elements comprising: a flexible cloud platform having a plurality of virtual machines, wherein each of the virtual machines has at least one fxCloudApp; and a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp, wherein the at least one fxDeviceApp and fxCloudApp are in secure communication to form a distributed application.

[0006] Another aspect of the disclosure is a plurality of network elements in a packet system, said network elements comprising: a flexible cloud platform having a plurality of zones, wherein each of the plurality of zones has at least one fxCloudApp; and a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp, wherein the at least one fxDeviceApp and fxCloudApp are in secure communication to form a distributed application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Figure 1 demonstrates the network architecture of a 4G system.

[0008] Figure 2 illustrates a network element (or system) upon which the Distributed Software Defined Network operates.

[0009] Figure 3 illustrates a Distributed Software Defined Network in accordance with the present disclosure.

[0010] Figure 4 is an exemplary view of the software layers in the Flexible Operation System (fxOS) Architecture.

[0011] Figure 5 illustrates the FastPath architecture.

[0012] Figure 6 presents a potential software layer of the fxOS co-existing alongside a general purpose OS.

[0013] Figure 7 is an exemplary view of the fxStore Portal.

[0014] Figure 8 demonstrates a shared fxCloud deployment.

[0015] Figure 9 demonstrates a Zoned Deployment.

[0016] Figure 10 demonstrates another deployment option as applied to cellular networks (especially Long Term Evolution (LTE)).

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0180

[0017] Figure 11 shows an example of a signaling flow chart for the fxDevice Commissioning Procedure.

[0018] Figure 12 shows an example procedure for Application Provisioning.

[0019] Figure 13 presents the state machine for the application upgrade process.

[0020] Figures 14A-14C are examples of the Distributed Switch Controller during Upgrade.

[0021] Figure 15 shows how fxVF in the fxCloud and fxDevice.

[0022] Figure 16 demonstrates the logical interfaces between the platform resources and applications.

[0023] Figures 17A and 17B demonstrate example procedures for cloud breathing procedures.

DETAILED DESCRIPTION

[0024] The Distributed Software Defined Network (dSDN) disclosed herein is an architecture that enables secure and flexible programmability across a network with full lifecycle management of services and infrastructure applications (fxDeviceApp). The dSDN also harmonizes application deployment across the network independent of the hardware vendor. As a result, the dSDN simplifies the network deployment lifecycle from concept to design to implementation to decommissioning.

DEFINITIONS, ACRONYMS & ABBREVIATIONS

[0025] The following terms, acronyms, abbreviations and descriptions are explained below and are used throughout the detailed description of the dSDN:

TERM	DESCRIPTION
2G	2nd Generation Cellular Technology
3G	3Rd Generation Cellular Technology
4G	4th Generation Cellular Technology
AP	WiFi Access Point
API	Application Programming Interface

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0181

AR	Application Repository
ASIC	Application Specification Integrated Circuit
AuC	Authentication Center
BS	Base Station
BSC	Base Station Controller
BTS	Base Station
CDN	Content Distribution Network
CN	Core Network
CNIE	Core Network Internet Edge
CPU	Central Processing Unit
dAP	Distributed Application Package
dApp	Distributed App
dCP	Distributed Content Service
dNS	Distributed Notification Service
DNS	Domain Name System
DPDK	Data Plane Development Kit
DPI	Deep Packet Inspection
dRS	Distributed Resource Service
DSCP	Differentiated Service Code Point (QoS)
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
EIR	Equipment Identity Register
eNB	Enhanced Node B (4G)
FIB	Flow Information Base
FQDN	Fully Qualified Domain Name
fxApp	FleXible Application, which has a fxDeviceApp and fxCloudApp component. fxDeviceApp and fxCloudApp of one application represent fxApp.
fxDeviceApp	FleXible Application which runs on fxDevice
fxBS	FleXible Base Station that is a BS that is powered by fxOS (an example of an fxDevice)

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0182

fxCloud	FleXible Cloud that interacts with fxDevice and on other network elements on the northbound interface
fxCloudApp	FleXible Cloud Application that is the application that runs on fxCloud and can be associated to one or more fxDeviceApp in fxDevice
fxManager	FleXible Manager that manager the fxOS lifecycle (provisioning, usage, de-provisioning)
fxOS	FleXible Operation System which run on the fxDevice as the OS and firmware
fxSDK:	FleXible Software Development Kit
fxSimulator	FleXible Simulator
fxStore	FleXible Store that presents fxApp to the network
administratorsGERAN	GPRS Edge Radio Access Network
GGSN	Gateway GPRS Support Node (used in core network of 2G/3G systems)
GMSC	Gateway Mobile Switching Center
GPRS	General Packet Radio Service
GPS	Global Positioning System
GTP	GPRS Tunneling Protocol
GW	Gateway
HAL	Hardware Abstraction Layer
HD	High Definition
HLR	Home Location Register
HSS	Home Subscriber Server
IAC	Inter Application Communication

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0183

IMS	IP Multimedia Sub-system
IMSI	International Mobile Subscriber Identifier
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
JVM	Java Virtual Machine
LA	Location Area
LAN	Local Area Network
LSP	Label Switching Protocol
M2M	Machine to Machine
MAC	Medium Access Control
MCP	Multi Chip Packaging
MIMO	Multi Input Multi Output
MME	Mobility Management Entity (used in core network of LTE)
MNO	Mobile Operator Network
MPLS	Multi Protocol Label Switching
MSC	Mobile Switching Center
MSISDN	Mobile Station
NAS	Non-Access Stratum
NB	Node B (3G)
NFC	Near Field Communication
NFV	Network Function Virtualization
NOIT	Network Operations & Information Technology; this term network broadly includes carriers, service providers, enterprise networks, and designated/contacted 3rd party administrators.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0184

OS	Operating System
OTT	Over the Top
P-GW	Packet Gateway
PaaS	Platform-as-a-Service
PBX	Private Branch exchange (telephony)
PDG	Packet Data Gateway
PDN	Packet Data Network
PHY	Physical Layer
PLMN	Public Land Mobile Network
PnP	Plug and Play
PP	Packet Processor
PR	Platform Resources
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RA	Routing Area
RAN	Radio Access Network of cellular networks
RAPI	Resources API
RFID	Radio Frequency Identifier
RNC	Radio Network Controller
S-GW	Serving Gateway
S1-AP	S1 Interface Application Protocol (an LTE protocol between MME and eNodeB)
SaaS	Software-as-a-Software
SDK	Software Development Kit
SDN	Software Defined Network
SGSN	Serving GPRS Support Node (used in core

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0185

	network of 2G/3G systems)
SIM	Subscriber Identity Module
SiP	System in a Package
SMS	Short Message Service
SoC	System on a Chip
SON	Self-Optimizing Network
SSDP	Simple Service Discovery Protocol
TA	Target Area
TNE	Test Network Environment
TTM	Time to Market
UMTS	Universal Mobile Telecommunication System
UPnP	Universal Plug and Play
URA	UMTS Routing Area
URI	Uniform Resource Identifier
UTRAN	UMTS Terrestrial Radio Access Network
VLR	Visitor Location Register
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WLAN	Wireless Local Area Network
XMPP	Extensible Messaging and Presence Protocol

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0186

[0026] Database. One or more large structured sets of persistent data maintained upon a computer system organized and structured according to a software system defining rules for organization as well as responding to queries to read, write, or modify data as well as provide statistical information regarding the contained data. As used in this disclosure in describing the dSDN, a database may be either a single unified system or a distributed system wherein certain database elements are located upon different systems or servers which may be in different physical locations, acting in harmony to appear as one unified database. Where databases are described, it will be understood by one of ordinary skill in the art that (i) alternative database structures to those described may be readily employed, and (ii) other memory structures besides databases may be readily employed. Any illustrations or descriptions of any sample databases presented herein are illustrative arrangements for stored representations of information. The database formats may include relational databases, object-based models and/or distributed databases which could be used to store and manipulate the data types described herein. Likewise, object methods or behaviors of a database can be used to implement various processes, such as the described herein. In addition, the databases may, in a known manner, be stored locally or remotely from a device which accesses data in such a database.

[0027] The term "determining" and grammatical variants thereof (e.g., to determine a price, determining a value, determine an object which meets a certain criterion) is used in an extremely broad sense. The term "determining" encompasses a wide variety of actions and therefore "determining" can include calculating, computing, processing, deriving, investigating, looking up (e.g., looking up in a table, a database or another data structure), ascertaining and the like. Also, "determining" can include receiving (e.g., receiving information), accessing (e.g., accessing data in a memory) and the like. Also, "determining" can include resolving, selecting, choosing, establishing, and the like. The term "determining" does not imply certainty or absolute precision, and therefore "determining" can include estimating, extrapolating, predicting, guessing and the like. The term "determining" does not imply that mathematical processing must be performed, and does not imply that numerical methods must be used, and does not imply that an algorithm or process is used.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0187

[0028] The functionality and/or the features of a single device that is described may be alternatively embodied by one or more other devices which are described but are not explicitly described as having such functionality/features. Thus, other embodiments need not include the described device itself, but rather can include the one or more other devices which would, in those other embodiments, have such functionality/features.

[0029] Devices that are described as in "communication" with each other or "coupled" to each other need not be in continuous communication with each other or in direct physical contact, unless expressly specified otherwise. On the contrary, such devices need only transmit to each other as necessary or desirable, and may actually refrain from exchanging data most of the time. For example, a machine in communication with or coupled with another machine via the Internet may not transmit data to the other machine for long period of time (e.g. weeks at a time). In addition, devices that are in communication with or coupled with each other may communicate directly or indirectly through one or more intermediaries.

[0030] Although process (or method) steps may be described or claimed in a particular sequential order, such processes may be configured to work in different orders. Further, some steps may be performed simultaneously despite being described or implied as occurring non-simultaneously (e.g., because one step is described after the other step) unless specifically indicated. Moreover, the illustration of a process by its depiction in a drawing does not imply that the illustrated process is exclusive of other variations and modifications thereto, does not imply that the illustrated process or any of its steps are necessary to the embodiment(s), and does not imply that the illustrated process is preferred. Where a process is described in an embodiment the process may operate without any user intervention.

[0031] It will be readily apparent to one of ordinary skill in the art that the various processes of the dSDN described herein may be implemented by, e.g., appropriately programmed general purpose computer(s), special purpose computer(s) and computing device(s). Typically a processor (e.g., one or more microprocessors, one or more microcontrollers, one or more digital signal processors) will receive instructions (e.g., from a memory or like device), and execute those instructions,

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0188

thereby performing one or more processes defined by those instructions. Instructions may be embodied in, e.g., one or more computer programs, one or more scripts.

[0032] A "processor" means one or more microprocessors, central processing units (CPUs), computing devices, controllers, microcontrollers, digital signal processors, or like devices or any combination thereof, regardless of the architecture (e.g., chip-level multiprocessing/multi-core, Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC), Microprocessor without Interlocked Pipeline Stages, pipelining configuration, or simultaneous multithreading).

[0033] Further, programs that implement methods described herein may be stored and transmitted using a variety of media (e.g., computer readable media) in a number of manners. In some embodiments, hard-wired circuitry or custom hardware may be used in place of, or in combination with, some or all of the software instructions that can implement the processes of various embodiments. Thus, various combinations of hardware and software may be used instead of software only to implement the embodiments.

[0034] The term "non-transitory computer readable medium" in this disclosure refers to any medium, a plurality of the same, or a combination of different media, that participate in providing data (e.g., instructions, data structures) which may be read by a computer, a processor or a like device. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media may include, for example, optical or magnetic disks and other persistent memory. Volatile media may include dynamic random access memory (DRAM), which typically constitutes the main memory. Transmission media may include coaxial cables, copper wire and fiber optics, including the wires that comprise a system bus coupled to the processor. Transmission media may include or convey acoustic waves, light waves and electromagnetic emissions, such as those generated during radio frequency (RF) and infrared (IR) data communications. Common forms of non-transitory computer-readable media in which the dSDN may be implemented include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, any other magnetic medium, a CD-ROM, DVD, any other optical medium, punch cards, paper

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0189

tape, any other physical medium with patterns of holes, a Random Access Memory (RAM), a programmable read only memory (PROM), an erasable programmable read only memory (EPROM), a flash electrically erasable programmable read only memory (FLASH-EEPROM), any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0035] Various forms of computer readable media in this disclosure may be involved in carrying data (e.g. sequences of instructions) to a processor. For example, data may be (i) delivered from RAM to a processor; (ii) carried over a wireless transmission medium; (iii) formatted and/or transmitted according to numerous formats, standards or protocols, such as Ethernet (or IEEE 802.3), Bluetooth, and Transmission Control Protocol/Internet Protocol ("TCP/IP"), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), and 2G/3G; and/or (iv) encrypted to ensure privacy or prevent fraud in any of a variety of ways well known in the art.

[0036] Thus in some embodiments in this disclosure a description of a process may be a description of a non-transitory computer-readable medium storing a program for performing the process. The computer readable medium may store (in any appropriate format) those program elements which are appropriate to perform the method.

[0037] In an embodiment, a server computer, network element or centralized authority may not be necessary or desirable. For example, an embodiment may be practiced on one or more devices without a central authority. In such an embodiment, any functions described herein as performed by the server computer or data described as stored on the server computer may instead be performed by or stored on one or more such devices.

[0038] In interpreting the present application (which includes the claims), one of ordinary skill in the art shall refer to the prosecution history of the present application, but not to the prosecution history of any other patent or patent application, regardless of whether there are other patent applications that are considered related to the present application, and regardless of whether there are other patent applications that share a claim of priority with the present application.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0190

[0039] Numerous embodiments are described in the present application, and are presented for illustrative purposes only. The described embodiments are not, and are not intended to be, limiting in any sense. One of ordinary skill in the art will recognize that the disclosed embodiment(s) may be practiced with various modifications and alterations, such as structural, logical, software, and electrical modifications. Although particular features of the disclosed invention(s) may be described with reference to one or more particular embodiments and/or drawings, it should be understood that such features are not limited to usage in the one or more particular embodiments or drawings with reference to which they are described, unless expressly specified otherwise.

[0040] The present disclosure is not a literal description of all embodiments of the invention(s). Also, the present disclosure is not a listing of features of the invention(s) which must be present in all embodiments. A description of an embodiment with several components or features does not imply that all or even any of such components/features are required. On the contrary, a variety of optional components are described to illustrate the wide variety of possible embodiments of the present invention(s).

[0041] The following description has been presented for purposes of illustration and description and is not intended to be exhaustive or to limit the embodiments to the precise form disclosed. Many modifications and variations are possible in light of the teachings disclosed herein.

[0042] The embodiments were chosen and described to explain principles of operation and their practical applications. However, the scope of the invention is to be defined by the claims.

[0043] The major technical reason for inflexibility in network infrastructure has been related mostly to the proprietary hardware and rigid software architecture in the existing network elements (routers, switches, gateways, cellular base-stations, WiFi access points, etc.). With recent advancements in semiconductor device manufacturing, there is no longer a cost barrier to add computing power to many electronics in the market including networking infrastructure. The addition of such compute capabilities would open up new opportunities for programmable platforms in the network and ultimately creating more flexible network architecture

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0191

and business models.

[0044] The Distributed Software Defined Network (dSDN) disclosed herein is an end-to-end architecture that enables secure and flexible programmability across a network with full lifecycle management of services and applications (fxApp). The dSDN also harmonizes fxApp deployment across the network independent of the hardware vendor. As a result, the dSDN simplifies the network deployment lifecycle from concept to design to implementation to decommissioning. In this disclosure, the dSDN is applied to the wireless networks as an exemplary embodiment. However, the dSDN is not limited to wireless networks embodiments and it could be applied to many other network types including enterprise, wireline service providers, data centers, and Internet of Things (IoT).

[0045] Figure 1 demonstrates the network architecture of a 4G system 100. (In alternative embodiments, the system and method disclosed may also be implemented in a 2G/3G network as well). In a cellular network, the base station or BS (denoted as BTS in 2G, NodeB or NB in 3G, and eNodeB or eNB in 4G systems) is responsible for termination of a wireless link (air-interface) to the mobile device (also known as the User Equipment (UE)). The BS then transports the user data to the core network and communicates control-signaling messages with the core network. In a typical cellular network, there are thousands of BS's covering nations. Each BS covers an area known as a cell. As the demand for mobile Internet expands, the mobile network operator (MNO) needs to deploy smaller cells so it can reuse the frequency more often.

[0046] A BS is made up of the following main functional blocks: 1) radio frequency (RF) front end; 2) clock; 3) baseband; 4) power manager; and 5) central processing unit (CPU). Some BS's also have a dedicated packet processor (PP) to accelerate packet processing in hardware. BS vendors may also use custom or merchant System-on-a-Chip (SoC) or Multi-Chip-Packaging (MCP) solutions to combine various functions into a single chip (e.g., PP, CPU, and baseband).

[0047] Even though there has been attempts to open up various interfaces in the BS (such as the Open BS Architecture Initiative that defines interfaces between these functional modules in the BS), there has been very little to no efforts to unify the

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0192

programmability of the BS itself. As a result, the MNOs suffer from the following difficulties. First, the core network Internet edge (CNIE), Packet Data Gateway (PDG) and the surrounding functions are becoming extremely complicated, non-scalable, and expensive. Second, each MNO needs certain customization and feature sets. Currently, they depend on their vendors for these customizations, which could cost the MNO tremendously both financially and in regards to time-to-market (TTM); hence, hindering innovation. In addition, usually these features are put into standards or in the vendor's feature set, thus eliminating the MNO differentiation against the other MNOs. Third, if the MNO is multi-vendor in their CNIE, coordinating all the vendors to implement the same features is usually a difficult and a time-consuming effort. As a result, more add-on network appliances are introduced into the CNIE which adds to the network management complexity. Fourth, as a BS becomes Internet Protocol (IP) based, there is more visibility into the user traffic types and new innovative opportunities are missed for creative traffic shaping features, backhaul bandwidth optimization, prioritization at radio edge, power management algorithms, etc. Fifth, many features implemented in CNIE present suboptimal performance. For example, the filtering enforcement at CNIE is fundamentally inefficient since the packets have to travel all the way through the expensive air-interface, the backhaul, various other core network elements and transport networks to get to the CNIE. Sixth, the Radio Access Network (RAN) deployments are a very expensive endeavor for MNOs. The current rigid BS designs limit innovation and force the MNOs to undergo major upgrades every few years.

[0048] The problems stated above apply to many different types of networks even though here they are presented in cellular networks as a focus. In the present disclosure, the dSDN exhibits a new paradigm in the software programmability of networks that would address the problems above and enable many more advanced features.

[0049] The dSDN may be located at a network element (or system) 200 which is shown in detail in Figure 2 or in alternative embodiments the functions of the dSDN may be divided among a plurality of network elements (or systems) which are similar in structure to network element 200. Each network element 200 may comprise one or

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0193

more system control logic 202 coupled with at least one or a plurality of processor(s) 204, system memory 206, a network interface 208 (including a transceiver 208a), and input/output (I/O) devices 210. The processor(s) 204 may include one or more single-core or multi-core processors. The processor(s) 204 may include any combination of general-purpose processors and dedicated processors (e.g., graphics processors, application processors, baseband processors, etc.). System control logic 202 for one embodiment may include any suitable interface controllers to provide for any suitable interface to at least one of the processor(s) 204 and/or to any suitable device or component in a packet network in communication with system control logic 202. System control logic 202 for one embodiment may include one or more memory controller(s) to provide an interface to system memory 206. System memory 206 may be used to load and store data and/or instructions, for example, for network element 200. System memory 206 for one embodiment may include any suitable volatile memory, such as suitable dynamic random access memory (DRAM), for example. System memory 206 may also include non-volatile memory including one or more tangible, non-transitory computer-readable media used to store data and/or instructions, for example, such as the embodiments described herein with regard to the dSDN. The non-volatile memory may include flash memory, for example, and/or may include any suitable non-volatile storage device(s), such as one or more hard disk drive(s) (HDD(s)), one or more compact disk (CD) drive(s), and/or one or more digital versatile disk (DVD) drive(s). The system memory 206 may include a storage resource physically part of a device on which the network element 200 is installed or it may be accessible by, but not necessarily a part of, the device. For example, the system memory 206 may be accessed over a network via the network interface 208 and/or over Input/Output (I/O) devices 210. Network interface 208 may include a transceiver 208a to provide a radio interface for network element 200 to communicate over one or more network(s) and/or with any other suitable device. Network interface 208 may include any suitable hardware and/or firmware. The network interface 208 may include a plurality of antennas to provide a multiple input, multiple output radio interface. Network interface 208 for one embodiment may include, for example, a wired network adapter, a wireless network adapter, a telephone modem, and/or a wireless modem. For one

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0194

embodiment, at least one of the processor(s) 204 may be packaged together with logic for one or more controller(s) of system control logic 202. For one embodiment, at least one of the processor(s) 204 may be integrated on the same die with logic for one or more controller(s) of system control logic 202. In various embodiments, the I/O devices 210 may include user interfaces designed to enable user interaction with the network element or system 200, peripheral component interfaces designed to enable peripheral component interaction with the network element or system 200, and/or sensors designed to determine environmental conditions and/or location information related to the network element or system 200. In various embodiments, the peripheral component interfaces may include, but are not limited to, a non-volatile memory port, a universal serial bus (USB) port, an audio jack, and a power supply interface.

dSDN Architecture

[0050] In the dSDN architecture, the network features are virtualized and may be distributed across separate network elements that cooperate together to create an advanced, programmable, and scalable network. In an alternative embodiment, the network features may be located at a single network element. A high-level overview of a dSDN system 300 is depicted in Figure 3. The dSDN system 300 consists of a flexible network device (fxDevice) 302, a flexible cloud platform (fxCloud) 304, an application management portal (fxManager) 306, and an infrastructure application market place (fxStore) 308. The fxDevice 302 is a network device that is powered by a sandboxing operating system named flexible operating system (fxOS) 302a. The sandboxing operating system ensures each application runs as a dedicated process in sure isolation from the other applications. The fxOS 302a may be built based on an existing sandboxed operating system (OS) (e.g., Android®) by extending several aspects of such an OS with routing/networking layers and supports for operating as the wireless infrastructure. The fxDevice 302 may host several independent and securely isolated applications (on top of fxOS 302a) named here fxDeviceApp 302b.

[0051] In the dSDN system, fxDeviceApp 302b may have a sister app in the backend cloud infrastructure (i.e., flexible cloud platform 304) referenced here as fxCloudApp 304a. The fxCloudApp 304a in the cloud is paired with its fxDeviceApp 302b in the fxDevice 302. The fxCloudApp 304a and the fxDeviceApp 302b

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0195

collectively form a distributed application (dApp or fxApp). In this description, when fxDeviceApp is referenced it shall mean any application that may have software components in the fxDevice, fxCloud, or both. In general, an fxApp package may include the following software components:

fxDeviceApp binary;

fxCloudApp binary;

Manifest files; and

Signatures.

[0052] It is important to note that the fxDeviceApp 302b and fxCloudApp 304a could use any protocol to communicate with each other and there is no need for standardizing this communication allowing for ultimate freedom for the developers. This allows for the system to operate in a loosely coupled autonomous fashion allowing for asynchronous communication in a distributed fashion.

[0053] The fxApps lifecycle is managed via fxManager 306. The fxManager 306 presents a user-friendly portal to the network administrator (admin) to discover, test prior to deployment, provision, and deprovision dApps. Using fxManager 306, the admin may discover new fxApps from the fxStore 308, which presents all the tested and certified vApps as well as showing the supported fxOS version, support hardware platforms, and other information such as reviews and number of commercial deployments by all NOIT customers.

fxOS

[0054] fxDevice 302 is defined as any networking equipment embedded with a special flexible operating system (fxOS) 302a. The purpose of fxOS 302a is to enable a carrier-class programmable platform. The fxOS 302a is configured to do, but is not limited to, the following. First, since fxOS 302a is designed for network equipment, it allows for efficient execution and running all the time. A plurality of power save modes may be implemented to ensure energy saving schemes without hindering the functionality of the fxDevice 302. Second, it enables simultaneous execution of several flexible applications (fxDeviceApp 302b) on the fxDevice 302. Third, it allows for application of a legacy compiled binary or byte-code of another language (for example, Java or Android®) in addition to fxDeviceApps 302b. Fourth, it allows

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0196

FastPath processing for data plane packets. The packet forwarding in this case separated from the control plane operations. The FastPath engine allows integration with specialized hardware accelerators or packet processors (PP). The packet forwarding may be implemented with software/kernel enhancements all in software. Another FastPath implementation may use both software/kernel enhancement and hardware packet processors (PP). Fifth, the fxOS 302a is capable of Deep Packet Inspection (DPI) and identifying all the flows and their related protocols, services and present this information in a format useful to application developers. The fxOS 302a can act on the data flows as the directed by the application developer. Sixth, applications are securely isolated from each other and to communicate with them is completely configurable by policies defined by the admin. Seventh, resource utilization of each application is policed such that an application may not be able to exceed its usage. For example, an application is pre-allocated the memory size, cache/storage space, and CPU cycles. Eighth, the applications may be signed with unique certificate security keys. The security keys and certificates may be allocated and/or signed by a certificate authority. The fxDevice 302 may be protected by validating signed applications to run or to install in the device. Ninth, it allows hot upgrade of the software and applications on the platform with little to no interruption to the operational aspect of the platform and its applications. Tenth, the fxOS 302a may boot by using the secure code from the network servers (cloud-booting). This would ensure tamper-proof software foundation for the fxDevice 302. Eleventh, there may be a secure and programmable firewall in the fxOS 302a. This embedded firewall protects against attacks to the fxDevice 302 and network attached to the network. The fxOS 302a firewall may be configured by the developer and it does allow replacing of attack detection algorithms (e.g., man-in-the-middle detection algorithm). The fxOS 302a may allow secure caching of user data, web pages, and media files (video and audio) in the fxDevice 302. The secure connections to the cloud (fxCloud 304) can enable sharing and control of the caching between the fxCloud 304, fxDevice 302, or the rest of the network. Twelfth, the fxOS 302a can support multi-threading and multi-core CPUs. This feature may dynamically distribute the threads to various CPU core without developers assistance.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0197

[0055] The fxOS 302a allows secure connection to the fxCloud 304. This secure connection allows communication between fxDeviceApps 302b in the fxDevice 302 and fxCloudApp 304a in the fxCloud 304. The fxOS 302a controls the access of applications to platform APIs. An application is categorized into an Access Level (AL) based on which the fxOS 302a decides that the application does not exceed its AL and access APIs that it is not authorized. The fxCloud 304 and the fxOS 302a create a virtual fabric (fxVF) for messaging between applications. The actual message can travel directly between two fxDevices or it may traverse the fxCloud 304. The messaging complexity may be abstract for the developer.

[0056] Figure 4 demonstrates an examplary view of the software layers in the fxOS 302a. Each layer of this software stack is modified and redefined to optimize for embedded networking equipment of wireless and wireline nature. Starting from the bottom of the stack, the kernel also connects to the hardware resources via the device drivers. Examples of such resources are wireless baseband SoC 402, hardware packet processor 404, Wide Area Network (WAN) uplink interface 406, cache and storage 408, and multi-core CPU 410. The OS kernel 412 is enhanced with the FastPath function 414 to accelerate packet processing (by help of hardware features such as Intel's Data Plane Development Kit (DPDK) or pure software enhancements) and other related data path functions. The enhanced kernel with FastPath 412, 414 would enable visibility into the data flows, sniffing the Non-Access Stratum (NAS) signaling (which is the signaling between the UE and the Core Network and the RAN is supposed to be transparent to), policy enforcements, regular packet forwarding functions, collection of flow statistics into Flow Information Base (FIB), and encryption/decryption functions.

[0057] Above the kernel, there are several tools 416, native daemons 418, native libraries 420 and Hardware Abstraction Layers (HAL) 422. These tools abstract out the hardware dependencies for the upper layers and programmers. The Runtime 424 is an embedded virtual machine capable of securely isolating and executing the applications. System Services 426 are a set of services always running and available to the developers (e.g., timing and messaging services).

[0058] fx-Framework Library 428 consists of several Frameworks. The new

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0198

frameworks are the library extensions introduced explicitly for the dSDN framework. Each of these frameworks adds a set of methods of (functions) and data structures for the following examples:

Wireless Manager 428a: manipulation, monitoring, and configuration of the wireless interfaces;

System Manager 428b: configuration, provisioning, and changing major settings of the fxDevice 302;

Flow Controller 428c: allows for manipulation of data path, tunneling, switching, and optionally coordinate this activity with fxCloud 304;

Energy Management 428d: enables fine-grained control over various elements in the fxDevice 302 that could be moved into various power stares, for example: ON, OFF, SLEEP, SUSPEND;

Big Data Manager 428e: manipulation, processing, and organization of data collected from various elements. This framework could enable close coordination between the fxDevice 302 and the fxCloud 304;

Content Manager 428f: for data sharing (database sharing) between the applications within the fxDevice 302 or the fxCloud 304 and between applications in fxDevice 302 and fxCloud 304. The Content Manager 428f may use URI to address the contents to be shared. The Content Manager 428f allows access to the application data according to the policy settings and permissions;

Policy Manager 428g: responsible for execution and implementation of polices set up by the admin. The application developers could use this framework to perform queries about the permissions, limitations, and rules;

Security Manager 428h: allows for access to the security protocol libraries and rewriting some of security algorithms such as the man-in-the-middle detection algorithm;

General Framework: general compute and logic building that may be

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0199

inherited from existing general OS frameworks.

[0059] Using the frameworks, there are potentially at least two types of application types that are possible. First, the System Native Applications 430 that are provided as the initial application load into the platform. These applications could be used by other applications by links or API. Second, the fxDeviceApps 302b that are written by third party developers or the NOIT.

[0060] Figure 5 further details the FastPath 414 architecture. In this approach, the FastPath 414 is divided into the Virtual Switch (data-plane) 502 and the Switch Controller (control-plane) 504. Such separation makes possible flexible deployment wherein the function of the Switch Controller 504 could move to the fxCloud 304 by an admin decision or by the dSDN system intelligence for example in the case of failure, redundancy, stateful reboot, or Hot Upgrade. The Virtual Switch 502 may include several functions and may interact with the Switch Controller 504 in the following exemplary ways:

Firewall 502a: implements the policies given to it by the firewall logic 504a in the fxOS Daemons.

Flow/NAS Discover 502b: continuously monitors data packets and discovers new flows and NAS signaling. It would then report it to the DPI Logic 504b where the flow intelligence resides and matching to meta-data happens. DPI Logic 504b in turn updates the Flow Information Base Information (FIB) 502c of the Virtual Switch 502.

FastPath & DPI Processing 502d: forwards, routes packets, and performs DPI functions as instructed by the FIB 502c. It also collects statistics of packets and reports them to the Monitoring Logic 504c.

Policy Enforcement 502e: this function enforces various policies on the traffic such as QoS marking according to the policies received from the Policy Logic 504d.

Shaper 502f: this function could shape (queue) the traffic according to the available bandwidth on the uplink or WAN interface.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0200

Virtual Fabric (fxVF) 504e: enables transparent switching of application and system messages of a dApp between the fxCloud 304 and the fxDevice 302 and between different dApps. The fxVF 504e is an abstraction layer that hides most of the complexities of messaging from the developers. The fxVF 504e uses the policies and firewall rules to enforce security and policies.

[0061] The fxOS 302a is mostly used for networking applications. There are some potential use cases where there is a need for a general purpose OS to run the legacy software applications with no need of recompilation. An example of a double OS situation may be where a digital signage (using a general purpose OS such as Android®) in a shopping mall also acts as a smart small-cell (using the fxOS 302a). In such a solution, any existing Android® application (for example) could be loaded in the fxDevice 302 and in parallel networking applications (fxDeviceApps 302b) could run on the same fxDevice 302.

[0062] Figure 6 presents a potential software layer of the fxOS 302a co-existing alongside a General Purpose OS 602. Another implementation may be a standalone fxOS without the company of another general OS. It is important to note that the shown co-existence allows reuse of the kernel so there would be no need to re-port the drivers of hardware resources twice.

fxCloud

[0063] The fxCloud 304 is an integral part of the dSDN framework. The fxCloud 304 connects to various parts of the fxDevice 302 to form a single virtual view of the system. Examples of the main fxCloud 304 responsibilities include the following. First, it interacts and manages all applications, firmware, and fxOS 302a in the fxDevice 302. Second, it secures isolation environments for distributed apps. Third, it directs the Virtual Machine lifecycle management including load-linked VM creations and destruction (Cloud Breathing). Fourth, the fxCloud 304 executes instructions defined by the fxManager 306. Fifth, it implements uniform policy (including security) distribution and execution across dSDN 300 (including the fxDevice 302, fxDeviceApps 302b, fxCloudApps 304a, and dApps). Sixth, Plug-and-Play commissioning of the fxDevice 302 introduction

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0201

and network expansion. Seventh, Distributed Resource Service (dRS) by which distributed applications can seamlessly communicate and message to each other using published APIs or proprietary methods within the pre-defined policies. Eighth, security violation discovery and alert system. Ninth, translation and/or exposure of Resources APIs (RAPI) to the systems outside the dSDN 300 within pre-defined policies. Tenth, fxCloud 304 in coordination with the fxDevice 302 creates a Distributed Notification Service (dNS) across the dSDN 300 by which the applications could be notified of an event to wakeup and/or respond to the event. Eleventh, fxCloud 304 in coordination with the fxDevice 302 creates a Distributed Content Management Service (dCMP) which provides a virtualized and distributed database system for applications to share application-specific or system-specific information with the pre-defined policies. Twelfth, the Switch Controller 504 function in the fxCloud 304 could take control of the controller function of the fxOS 302a in the fxDevice 302. This would allow controller switching from local to remote and back. Examples of such a configuration is a temporary reason where the fxOS Switch Controller 504 may not be available during reboot or software upgrade.

[0064] In the discussion below, these services and features are expanded upon.

fxDeviceApp-0 & fxCloudApp-0

[0065] The fxDevice 302 and fxCloud 304 have embedded System Applications categorized as fxDeviceApp-0 and fxCloudApp-0, respectively. The fxDeviceApp-0 is responsible for all default and core functional capabilities of the fxDevice 302 as listed below:

Firmware of the fxDevice;

Default communication protocols (e.g. LTE S1-AP, baseband software, security protocols, GTP, DNS, tunneling protocols);

Message communication protocol between the fxDevice and fxCloud;

Initial fxDevice settings;

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0202

Boot Loader;

fxDevice X.509 digital certificate;

OS and Kernel upgrades;

Upgrade and commissioning agents for the fxDeviceApp-0 upgrades and related procedures;

Establishing and initiating the message communication protocol between the fxDevice and fxCloud;

General fxDeviceApp install and upgrades;

Access to fxDevice hardware and software information.

The fxCloudApp-0 is responsible for all default and core functional capabilities of the fxCloud 304 such as:

Upgrade and commissioning agents for the fxCloudApp-0 upgrades and related procedures;

Hypervisor upgrades;

Virtual Machine lifecycle management (e.g. creation, suspension, destruction);

fxCloud X.509 digital certificate management;

OS and kernel upgrades;

General fxCloudApp install and upgrades;

Default communication protocols (e.g. security protocols, DNS, tunneling protocols).

Each software component can be managed and updated individually. To simplify the management of these software components, each software components can be sub-numbered such as fxDeviceApp-0.1, fxDeviceApp-0.2, fxCloudApp-0.1, fxCloudApp-0.2, etc.

fxManager

[0066] The network administrator may use the fxManager 306 to control much of the dSDN system 300. The fxManager portal could use HTML5, for example, as

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0203

a frontend technology. The major functions of the fxManager 306 may include the following.

fxDevice Management:

Commissioning (plug-and-play), de-commissioning, firmware/OS upgrading/downgrading;

Suspend/resume functions;

Unification of zones and partitions for example in case of network merger;

Detailed management of fxDevices (remote secure login) where the administrator may zoom into a particular or a group of fxDevices to monitor and set management alarms;

Security policy setting.

Application lifecycle management: fxManager 306 may work, coordinate, and/or perform these tasks in conjunctions with fxCloud 304 for:

Provisioning, de-provisioning, upgrade/downgrade applications. Some applications may only have a fxDevice component (Device-Centric Apps), may only have fxCloud component (Cloud-Centric Apps), or may have both fxDevice and fxCloud components (Hybrid Apps);

Creating a Test Network Environment (TNE) where the administrator can test certain configuration and applications without impacting the production networks;

Uploading applications developed by the NOIT or their partners into the Application Repository (AR). The AR is placeholder where the application packages are downloaded and stored for testing and verification prior to installation into the live dSDN system. An application could have several components (e.g. fxDeviceApp binaries, fxCloudApp signatures, manifest files, signatures) bundled together to form the Distributed Application Package (dAP);

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0204

Browsing through the certified applications in the fxStore and downloading them to the AR;

Application license accounting services by which a secure and accurate accounting of the installed applications are performed regularly. This inventory management function allows the administrator and the 3rd party ISV to account for live and active instants of the software in the network;

Application review tool where the network admin could use to review feedbacks and comments of other NOIT (maybe anonymous reviews) and could input his/her reviews for others (maybe anonymous reviews).

fxCloud Management:

Zoning and partitioning the network for administrative purposes. Zones and partitions could optionally put into different Virtual Machine (VM);

VM load management and Cloud Breathing policies;

The configuration conflict resolutions. Some configurations applied to the resources may conflict with each other (e.g., one firewall rule may state drop Netflix packets explicitly and another rule may state allowing Netflix explicitly). The fxCloud resource manager could monitor these configurations and discover the conflicts and signal those to the network administrator.

Security Management:

Security policy setting;

Inter-App communication setting;

Security of communication between fxDevice and fxCloud;

API Access Level (AL) definitions and (re-)assignment to the applications (permissions);

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0205

Risk Analysis of the applications;

Certificate and signature verification of the applications;

Content sharing between applications and associated policies.

[0067] An fxManager Portal per the above description may have an "Account" button which can be used to create admin users and their privileges. The fxManager Portal could also include a geographical map illustrating locations of fxDevices and their status. The fxManager 306 functions could be integrated into other network management tools. The fxManager 306 could potentially build the inventory of the network devices (fxDevices) showing all the available devices and statistics. As described above, the network administrator could optionally procure the applications a third party hosted application store (fxStore). The fxStore 308 could potentially categorize the applications several groups (as shown in Figure 7 which is an example view of the fxStore portal). The fxStore 308 could also show the reviews of the advertised applications. In order to maintain quality, the third party fxStore 308 may perform rigorous testing of the applications and accompany those results with the advertised application.

fxSDK

[0068] The flexible Software Development Kit (fxSDK) may include a development environment and other tools to facilitate development, testing, debugging and verifications of applications for dSDN environments. The fxSDK is unique in several aspects exemplified as follows. First, since the dSDN applications may have a device component (fxDevice 302) and cloud component (fxCloud 304), the fxSDK allows the developer to develop the components together to simplify the development and testing. Second, the dRMS offers various system level APIs with remote accessibility. In other words, the APIs in the fxDevice 302 could be accessed via the applications in the fxCloud (fxCloudApps) and vice versa. The fxSDK simplifies the usage system created APIs and application created APIs. Third, it is essential for the dSDN to ensure quality and security of applications in the network. The fxSDK may be accompanied with an extensive Risk Analysis Utility (RAU) which verifies all the APIs that are being used by the application and based on its potential danger to the

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0206

network, it would show the risk analysis and may make specific suggestions to reduce the risk. Once completed, the risk analysis result is included in the dAP for posting to the fxStore and consumed by the fxManager user (the admin). Fourth, the fxSDK would also ensure that the applications do not exceed their planned memory and resource usage. Fifth, the Integrated Development Environment (IDE) may provide various tools to actively demonstrate errors, defined APIs, security dangers, and excessive resource usage. Sixth, the IDE could also provide a flexible Simulator (fxSimulator) for the developer test the application in a network setup in presence of other network elements.

Deployment Examples

[0069] The dSDN 300 offers flexible deployment options. In this section, a few exemplary embodiments are presented.

[0070] Figure 8 demonstrates a shared cloud deployment where several NOITs (802, 804, 806) share an fxCloud 808. In order to ensure complete isolations, each NOIT may be assigned a separate Virtual Machine (VM) 802a, 804a, 806a. A good example of this deployment option may be when multiple service providers outsource their fxCloud deployment to a third party vendor that offers a SaaS/PaaS solution. It is important to note that in this option, each NOIT should have its own secure instance of the whole system. This option could potentially create a full outsourcing model for network operations including possibilities of seamless merging of networks of two or more NOITs.

[0071] Another deployment option may be a Zoned Deployment as shown in Figure 9. In this option, a large NOIT may partition their network into a plurality of zones (Zone #1, Zone#2,… Zone#n) for administrative, software compatibility grouping, or other reason. For example, it is possible that in a large network deployment, there would be different versions of fxOS with varying capabilities in the network (the older hardware may not support newer fxOS version). In this case, the service provider may decide to group the fxDevices per fxOS versions for management simplifications.

[0072] Figure 10 is another embodiment deployed in a Programmable & Virtualized Cellular Network. In this example, the dSDN is applied to the cellular

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0207

networks (e.g., LTE). As shown, the fxDevice 1002 here is the LTE eNB (4G base-station) and the fxCloud 1004 hosts the LTE core network (EPC) functions such as MME, S-GW, and P-GW in separate VMs. Within each EPC function, new features could be added using the dSDN framework. For example, a new application to optimize signaling for Machine-2-Machine (M2M) devices could be loaded into the fxDevice 1002 and the MME function in the fxCloud 1004.

PRINCIPAL PROCEDURES & SERVICES

[0073] The dSDN framework could offer extensive services and capabilities that would simplify the programmability of the network. In this section below, a set of main procedures and services are presented as further exemplary embodiments.

fxDevice Commissioning Provisioning

[0074] Figure 11 illustrates a signaling flow chart for a fxDevice Plug-and-Play Commissioning Procedure. This procedure enables fxDevice commissioning and initial setup to provide a seamless and Plug-and-Play (PnP) deployment. The fxDevice relies on the fxDeviceApp-0 and the fxCloudApp-0 to connect to the backend management system and perform the commissioning procedure. In step 1101, the fxDevice 1120 tries to discover the fxManager 1130 using the fxManager Fully Qualified Domain Name (FQDN). The fxDevice 1120 is configured to always look for the same FQDN. If the NOIT has a private cloud deployment, the DNS server in the NOIT will be configured to point to its own fxManager IP address. If the cloud is shared, the NOIT will configure their DNS server to point to the shared cloud IP address. In step 1102, the fxDevice 1120 receives the IP address of the fxManager 1130. In step 1103, the mutual authentication may be performed where the fxDevice 1120 authenticates the fxManager 1130 and vice versa. In step 1104, a secure tunnel may be created between the fxDevice 1120 and the fxManager 1130. In step 1105, the fxDevice 1120 may optionally request for the latest firmware/OS version. In this phase, the fxDevice 1120 informs the fxManager 1130 of its hardware information (e.g. vendor, model number, CPU model/speed, memory/cache size) and software information (e.g., current firmware version, current OS version, fxDeviceApp-0 version package stating versions of fxDeviceApp-0 components). In step 1106, the fxManager 1120 may send the latest firmware/OS version (or point

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0208

the fxDevice to the right URL to download the latest load). In step 1107, if a new firmware/OS load is downloaded and verified, the fxDevice 1120 installs it. In step 1108, the fxDevice 1120 requests the default settings and the default application package (the initial applications that the NOIT admin would like to install in fxDevice 1120 prior to other applications). In step 1109, the fxManager 1130 prepares the fxCloud 1140 for the new fxDevice 1120 integration into the network. In step 1110, once completed, the fxCloud 1140 informs the fxManager 1130 of its readiness. In step 1111, the fxManager 1120 sends all the default settings and the default application package to the fxDevice 1120. In step 1112, the fxDevice 1120 then sets up a secure connection to the fxCloud 1140 (more precisely the fxCloudApp-0). In step 1113, a commissioning completion message is set from the fxCloud 1140 to the fxManager 1130. In step 1114, the fxManager 1130 signal to the fxDevice 1120 the completion of the commissioning and starts th regular operation of the fxDevice 1120. At this phase, fxManager 1130 can optionally modify the settings in the security connections between fxDeviceApp-0 and fx-CloudApp-0 and between fxDeviceApp-0 and fxManager 1130.

Application Provisioning Procedure

[0075] The Application Provisioning Procedure enables provisioning and de-provisioning of applications across the dSDN system. For simplicity, the term provisioning in this disclosure is used to present all similar procedures of provisioning and de-provisioning. The application provisioning procedures are usually triggered by the network administrator and are orchestrated by the fxManager, which works in collaboration with the fxDeviceApp-0 and fxCloudApp-0.

[0076] Figure 12 shows an example procedure for Application (dApp) Provisioning. In step 1201, the network admin identifies the applications that would need to be installed in the dSDN system. These applications are already downloaded (from the fxStore or other sources) into the Application Repository (AR). Prior to installation in the live network, the authorized network admin 1220 has probably performed rigorous testing of the application in the test network. In step 1202, the fxManager 1230 performs a detailed analysis on the potential risks that this application may

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0209

create on various parts of the network and other applications. The analysis tool uses the secure manifest file and other information to discover, for example, all the APIs used by the application, the required access-level authorization level for each of those APIs, requirements on the fxVF (inter-app, intra-app communication needs), and usage of platform resources including conflict discovery and resolutions. In step 1203, the admin defines the deployment scope that the application would be applied. Examples of deployment scopes could be a given zip code, a city, an administrative domain (zone), a shopping mall, an office building, or manual handpicking of sites or fxDevices 1240. The fxManager 1230 verifies compatibility of the OS version and resouces requirements of the application and the fxDevices 1240. If the fxManager 1230 discovers incompatibility, it informs the admin and request adjustment to the deployment scope or proposes upgrading incompatible fxDevice if possible. In step 1204, the admin 1220 initiates the installation process. In step 1205, the fxManager 1230 unpacks the dApp and if there is fxDeviceApp component, it would prepare it for submission to the fxDeviceApp-0 of fxDevices in the deployment scope. In step 1206, installation commands are sent to the fxDeviceApp-0s in a secure connection already set up as part of the fxDevice Commissioning. In step 1207, the fxDevice 1240 verifies the fxDeviceApp package and performs necessary integrity checks of the received software package. Once verified, the fxDeviceApp is installed and re-verified. In step 1208, the completion of successful install is sent to the fxManager 1230 by the fxDevice 1240. In step 1209, the fxManager 1230 prepares the fxCloudApp component of the dApp (if any). In step 1210, installation commands are sent to the fxCloudApp-0s in a existing secure connection. In step 1211, the fxCloud 1250 verifies the fxCloudApp package and performs necessary integrity checks of the received software package. Once verified, the fxCloudApp is installed and re-verified. In step 1212, the completion of successful install is sent to the fxManager 1230 by the fxCloud 1250. In step 1213, the fxManager 1230 informs the admin 1220 of the successful installation process.

Hot Upgrade Procedure

[0077] The Hot Upgrade refers to a procedure by which the software upgrades

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0210

on the system have no or minimal implication on the functionalities offered by the software component subject to the upgrade process. Herein are presented two types of software upgrades. For simplicity, the use of the term upgraded in this disclosure represents both upgrade and downgrade since they both use identical processes. First, upgrade of applications on the fxDevice (i.e., fxDeviceApp) and fxCloud (i.e., fxCloudApp). Second, upgrade of core functions in the fxDevice (aka fxDeviceApp-0) or fxCloud (aka fxCloudApp-0).

[0078] The upgrade procedures for fxDevice and fxCloud are similar. Here, the focus is on the fxDevice upgrade process since it is technically more challenging due to stricter resource limitations in the fxDevice.

General Application Upgrade

[0079] The applications could be upgraded automatically or manually by the network admin. In either case, the installation of the upgrade follows a similar procedure as the Application Provisioning Procedure. Figure 13 presents the state of the machine for the application upgrade process. In state 1301, the old version of the application is still running. In Load state 1302: the new version of the application is downloaded in the local storage while the old version is still running. In Install state 1303: the new version is installed (in the memory) while the old version is still running. In Run state 1304: the new version is activated taking charge of all the related data and application state and the old version is deactivated while still remaining in the local storage. In Commit state 1305: the old version is fully removed after ensuring the new version is running properly as expected.

fxDeviceApp-0 Upgrade

[0080] The fxDeviceApp-0 is a collection of software components that perform the core functionalities of the fxDevice. As a result, this hot upgrade is quite challenging. The most important thing is to ensure that major functionalities of the fxDevice (such as packet forwarding) remain

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0211

intact during the upgrade process. The fxDeviceApp-0 upgrades may be categorized as:

- No OS Kernel upgrade; or
- OS Kernel upgrade.

No Kernel Upgrade

[0081] As described earlier, the packet forwarding function could be broken into the Virtual Switch and the Switch Controller. In order to maintain, the packet forwarding function, the Switch Controller function could be performed by another device or a second processor in the same device while the fxDevice is being upgraded. Figures 14A-14C demonstrate a few examples of distributed Switch Controller function during the upgrade. In some cases, it may be possible for the fxCloud to take over the Switch Controller function (Figure 14A), or a neighboring fxDevice (Figure 14B), or second CPU in the same fxDevice (Figure 14C). The actual software upgrade procedure would be similar to Application Provisioning Procedure and the state machines are the same as the one shown in Figure 13.

Kernel Upgrade

[0082] In this case, the packet forwarding process is mostly unavailable. Therefore, one potential solution might be for the fxDevice to redirect the entire bit stream to a neighboring fxDevice. In other words, the fxDevice would have just a bare minimum routing function working and the rest of the function would be performed by a neighboring fxDevice.

[0083] All the other upgrade procedures and state machines are similar to the ones performed for the No Kernel Upgrade process.

Virtual Fabric Service

[0084] The Virtual Fabric (fxVF) provides an abstraction layer for application to communicate with each other whether they are in the fxDevice or fxCloud. Various frameworks and services may use the fxVF service. Figure 15 illustrates fxVF 1502a, 1503a, 1504a in the fxDevices 1502 and 1503 and fxCloud 1504. The fxVF 1502a,

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0212

1503a, 1504a may use Extensible Messaging and Presence Protocol (XMPP) for this messaging as the default protocol. It is important that the developers may use their own communication protocols between the fxDeviceApps 1502b, 1503b and the associated fxCloudApp 1504b. fxVF 1502a, 1503a, 1504a provide a secure routing mechanism.

Intra-App Messaging

[0085] This is an example of where one application in the fxDevice communicates with the same instance of the app in another instance of the same application in another fxDevice. The actual messages could go directly between the fxDevice or via the fxCloud. As an example, in the case of mobile networks, this messaging could be used to transfer user specific context from one eNB to another as a user hands off to a new eNB. In LTE, Private Messages on X2 interface could be used for messaging between the eNBs that act as the fxDevices.

Inter-App Messaging

[0086] In some cases, different applications may need to communicate with each other via their published APIs. In this case, the security policies set up by the network administrator determines which applications could communicate with each other for what purpose. The fxVF follows the security policies determined by the network administrator for inter-application communications.

distributed Resources Service (dRS)

[0087] The resources could be platform resources or APIs offered by the applications. The fxCloud and fxDevice resources could include (as examples):

Firewall;
Data and statistics;
Storage (usually abundant at fxCloud);
Compute (usually abundant at fxCloud);
Load evaluator;
General settings;

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0213

Routing engine (usually relevant to the fxDevice);

Virtual Machine master controller (usually relevant to the fxCloud);

Power controller (usually relevant to the fxDevice); and

Wireless engine (usually relevant to the fxDevice).

[0088] An app (fxDeviceApp or fxCloudApp) could expose APIs to be used by the other apps. The inter-app communication is enabled by the fxVF where the policy and security provisions are enforced. The APIs exposed may be RESTful (representational state transfer) and could travel across physical network elements. Since the fxManager defines fxVF policies, the administrator could ultimately specify which APIs between which apps could communicate with each other. Figure 16 demonstrates the logical interfaces between the platform resources and applications. As shown in Figure 16, the distributed Resources Service (dRS) 1602, 1603 comprises software agents residing in the fxDevice 1604 and the fxCloud 1606. These dRS agents manage the access to the platform resolutions (including potential configuration conflict resolution) as well as facilitating inter-app APIs. The dRS 1602, 1603 allows or disallows access to platform resource or inter-app communication according to the policies defined by the network administrator using the fxManager. In summary, the dRS 1602, 1603 provides services to application developers for:

Exposing APIs to other applications;

Configuring and managing platform resource;

Policy enforcement and authorization of applications access to platform resource and other app's APIs; and

Policy conflict resolution.

distributed Content Service (dCS)

[0089] The distributed Content Service (dCS) allows the developers to seamlessly store and share the contents generated by one application with other applications and its associated application in the cloud. The dCS simplifies access to the data and brings in storage virtualization to the applications. In other words, the developers no longer would need to know where the data is actually stored (in

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0214

the cloud or on the device) and would be able to access them easily. The dCS implementation may use Virtual Fabric (fxVF) and distributed Resources Service (dRS).

distributed Notification Service (dNS)

[0090] The distributed Notification Service (dNS) is another potential tool for developers that could wake or ping an application when a particular event has occurred. For example, a load monitoring application could be notified when the CPU load on a particular fxDevice (or a target area) exceeds a certain threshold. In turn, such an exemplary application could make smart decisions on reducing the load on the CPU by forcing handoffs of users to neighboring cells (i.e. fxDevices). The dCS implementation may use Virtual Fabric (fxVF), distributed Resources Service (dRS) and distributed Content Service (dCS).

Cloud Breathing Procedures

[0091] In general, resources (compute, storage) at the cloud are more abundant. However, software licensing costs and other limitation may require smart management of resources. The Cloud Breathing, here, is defined as a mechanism where the cloud resources are automatically expanded as load increases on the system or reduced as the load decreases. This creates an automatic elasticity in the cloud dimensions. Figures 17A and 17B demonstrate procedures for the cloud breathing. Figure 17A is expansion and Figure 17B is reduction. The Load Controller in the dRS 1602, 1603 could be used to monitor the load (step 1702) on the existing VMs (e.g. CPU or memory utilization loads). The VM Master of the fxCloudApp-0 1608 could effectively act on the decisions made by the dRS's Load Controller. Again, the network administrator via the fxManager defines the policies and thresholds for such decisions.

Security Aspects

[0092] The security aspects of this solution are of outmost importance in order to ensure quality in the product networks.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0215

Identity

[0093] To create a security model, the dSDN network elements should have unique identities. The following identities may be defined:

fxDevice Identity (DID): this identity is unique globally and may be allocated at manufacturing and included in the X.509 digital certificate of the fxDevice;

fxCloud Identity (CID): this identity is unique to the NOIT domain and may be allocated by the network administrator and included in the X.509 digital certificate of the fxCloud;

fxDeviceApp Identity (DAID): this identity may be globally unique and may be allocated by the fxSDK at the point of code creation by a globally accessible server;

fxCloudApp Identity (CAID): this identity may be globally unique and may be allocated by the fxSDK at the point of code creation by a globally accessible server;

dApp Identity (AID): This identity may be globally unique and may be constructed by concatenation of DAID and CAID together;

Application Developer Identity (ADID): this is a globally unique identity allocated by a globally accessible server and may be included in the developer's X.509 digital certificate and is used to sign the final developed applications.

[0094] For addressability purposes, the identities may be presented in a URI format using the FQDN of the NOIT.

Access Level

[0095] The applications are given different access levels. The AL is used by the dRS to determine which APIs or class of APIs an application can access (e.g., a fxDeviceApp running on fxDevice should not be able normally to reboot the fxDevice). The required AL is generated by the fxSDK and included in the manifest files. It may also be published in the fxStore for that give application. A preset of ALs could be defined to categorize the applications security risks.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0216

Software Security Verification

[0096] It is important for the target network element to verify the authenticity and integrity loaded software prior to install. The uniquely defined security keys of the application developers sign the applications. The following rules may be followed to ensure software security verification:

Only the verified OS/Firmware software can install on verified hardware;

Only the verified OS/Firmware software can run on verified hardware (continuous verification of essential parts of the software);

Only the verified fxDeviceApp software can install on verified hardware;

Only the verified fxDeviceApp software can run on verified hardware (continuous verification of essential parts of the software).

[0097] Each fxDeviceApp is accompanied by a manifest that is generated by the Integrated Device Electronics (IDE) at the time of compilation of the fxDeviceApp. The manifest file specifies which libraries and frameworks the fxDeviceApp uses. The fxManager uses this information to determine the security risk of the fxDeviceApp as well as the required Access Level (AL) to run this fxDeviceApp. In order to ensure that manifest file is actually genuine manifestation of the fxDeviceApp and its integrity remains intact, a hashing algorithm could be used to generate a signature. An example is shown below:

Signature = HMAC-SHA256(K, Message);

where

Message = fxDeviceApp + Manifest

K = shared secret key between fxStore and fxManager.

[0098] The signature is generated by the IDE/fxSDK at the compile/build phase and is packaged with application binary and the manifest file. Once the distributed application package (dAP) is downloaded into the fxManager (Application Repository), the components of the signature, application binary, and manifest file are unpackaged. The fxManager then uses the same hashing algorithm to calculate

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0217

the signature as above using the pre-shared key (K). If the calculated signature and the unpackaged signature match, it would prove the authenticity of the manifest file and the integrity of both manifest file and the fxDeviceApp.

[0099] Code Generation Process (Compilation):

fxDeviceApp compiled→ Manifest file generation→ Signature calculation→ Create fxDeviceApp Package

[0100] Install Process (Consumption):

Un-package fxDeviceApp→ Signature calculation→ Signature comparison

[0101] This model works if the key K can be shared with two trusted parties (fxManager and fxStore). The manifest file that is generated by the fxSDK shall be secured by integrity checks to ensure it remains intact throughout the transaction (i.e. from compilation to consumption).

Communication Security

[0102] Security mechanisms built into the fxDeviceApp-0 and fxCloudApp-0 forces the fxDevice to only use the configured fxCloud servers for Software Security Verification, fxDeviceApp downloads, and fxDeviceApp-fxCloud communication. Optionally, the fxDeviceApp-fxCloud communication can be customized per application requirements. One implementation of this interface could use SPDY Protocol or per-application VPN.

[0103] For the purpose of inter-App communication, an Inter-App Communication (IAC) policy template is applied to an fxDeviceApp that defied the communication policies between the fxDeviceApps and fxCloudApps. The network administrator may use one of the pre-configured IAC policy templates (exemplified below):

Isolated: this type of applications cannot communicate with any other application in the same fxDevice and does not have any associated fxCloudApp;

Isolated-Extended: this type of fxDeviceApps cannot communicate with any other fxDeviceApps in the same fxDevice but could communicate with its associated fxCloudApp;

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0218

Private: a private fxDeviceApp in one fxDevice could communicate with same private fxDeviceApps in other fxDevice but does not have any associated fxCloudApp;

Private-Extended: a Private fxDeviceApps in one fxDevice could communicate with same private fxDeviceApp in other fxDevice and has its associated fxCloudApp;

Community: a group of fxDeviceApps that belong to the same community can communicate with each other;

Promiscuous: a Promiscuous app could communicate with any other Promiscuous application and Community application;

Custom: the network administrator could create custom IAC Policy templates and could include (as examples):

Template name;

Inherited pre-configured template if any;

The protocols allowed to use for communication;

The maximum data rate allowed for inter-app communication between a pair of fxDeviceApps or between an fxDeviceApps and an fxCloudApp;

Policies related to the other fxDeviceApps in the same fxDevice, in other fxDevice, associated fxCloudApp and unassociated fxCloudApps;

Policies related to direct fxDeviceApp to fxDeviceApp communication or relayed via fxCloud communication.

Secure Boot

[0104] The secure boot refers to a capability where the main boot code is in the fxCloud with the following exemplary procedure:

The Boot Loader (BL) in the fxDeviceApp-0 of the fxDevice looks for a particular DNS (DNSSEC) name of the fxCloud. The network administrator should define the DNS entry in their network. If there is a private fxCloud, the resolved domain name points to the fxCloud. If using shared cloud, the resolved

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0219

domain name points to the cloud point of presence;

A secure tunnel (TLS) is created between the BL in fxDevice and the fxCloud using pre-burned X.509 certificate;

BL identify itself to the fxCloud and include System Information;

fxCloud determines the appropriate boot file and points the BL to the exact address of the boot file in the fxCloud;

BL downloads the correct boot file;

BL runs Software Security Verification;

BL reboots the fxDevice and the system will be ready for operation.

Frameworks

[0105] The following describes an example of the API Framework for the dSDN. Some APIs are purely local to the fxDevice or the fxCloud and some may be extended from the fxDevice all the way to the fxCloud using dRS.

General Framework

[0106] This framework refers to general libraries and APIs inherited from the legacy OS. An example of the legacy OS may be Embedded Android®. This framework allows for general processing and may be used for algorithmic applications.

Wireless Framework

[0107] The Wireless Framework adds a set up functions related to managing and controlling the wireless module. Table 1 highlights some examples of methods/functions that may be available in the Wireless Framework.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0220

Table 1: Wireless Framework	
Method	Description
discoverNeighbors	Discovers wireless neighbors and returns their identities
retrieveNeighborInfo	Retrieves detail information of neighboring BS/AP information. This method may transparently call cloud servers to gather more information
measureInterference	Measures wireless interference observed on a frequency or a set of frequency channels
setWirelessConfig	Sets various parameters in the wireless module such as power, frequency channels, Multiple Input/Multiple Output (MIMO) antenna configuration, etc.
definePHYReceiverAlgorithm	Defines the physical layer receiver algorithm
connectedDevice	Returns details about the connected mobile devices (length of connection, data tx/rx) to a given wireless interface (channel, band, sector)
forceOffload	Offloads connected mobile devices to other access technologies (e.g. from cellular to WiFi)
homeAreaNetworking	Control of home automation wireless technologies such as ZigBee or Z-Wave
setNeighborCells	Sets neighboring cells information
forceHandoff	Handoffs connected mobile devices to the same access technology but different channel or frequency
topMovers	Identifies the mobile devices with highest degree of mobility

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0221

positionDevice	Accurate positioning of mobile devices which may include geo-fencing data
borrowSpectrum	Allows one BS to borrow spectrum from another
configureMACScheduler	Configures the MAC scheduler (for example prioritize a particular user)
loadBalance	Enables load balancing across radio channels
defineMACScheduler	Defines the MAC scheduler algorithm

Security Framework

[0108] The Security Framework allows the developers to use a specialized security functions/methods. Some of these new methods may use the legacy OS frameworks. Table 2 highlights some examples of methods/functions that may be available in the Security Framework.

Table 2: Security Framework

Method	Description
setDOSDetect	Detects DOS attacks on user plane as requested
configureFirewall	Provides firewall functions such as traffic filtering, permissions (this may use dRS)
defineDOSDetectAlgorithm	Allows replacing and redefining the DOS algorithm
encryptTraffic	Encrypts the content of a message or a class of traffic as requested
authenticateUser	Authenticate users as requested
integrityCheckContent	Verifies authenticity of a message using its checksum information
configuresIPS	Configures IPS functions on the user plane

FastPath Framework

[0109] The FastPath Framework allows the developers to identify and manipulate the data path with a DPI capability. Table 3 highlights some

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0222

examples of methods/functions that may be available in the FastPath framework.

Table 3: FastPath Framework

Method	Description
matchTraffic	Identifies and matches the traffic type requested by the developer
detectCongestion	Detects if there is congestion on the data path
identifyTopURLs	Identifies the most visited URLs by the users
detectLocalConversation	Detect conversations and packet routing between the end user on the same fxDevice
detectDevice	Detects traffic of a given mobile user device
redirectTraffic	Redirect data traffic to a given network element. The traffic may be identified with a user or a class of traffic
connectedDevices	Returns all the connected mobile devices to a given fxDevice
assignAddress	Allocates a particular IP address to the mobile device
setPriority	Set traffic priority to a class of traffic
markTraffic	Marks the user traffic with the desired QoS tags
setupTunnel	Sets up data tunnel from fxDevice to a destination
Adscript	Adds a JavaScript to an active transiting web page
detectIdenticalFlows	Identifies the identical flows coming from the WAN/uplink interface on the fxDevice
aggregateFlows	Enables aggregation of flows to save WAN/uplink utilization
routeTunnel	Routes a packet or a class of traffic onto an established tunnel
identifyTopDevices	Identifies users that generate most traffic load

Messaging Framework

[0110] The Messaging Framework allows the developers to send messaging between the applications residing in the fxDevice and fxCloud. Table 4 highlights some examples of methods/functions that may be available in the Messaging Framework.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0223

Table 4: Messaging Framework	
Method	Description
sendIntraApp	Sends an Intra-App message from one fxDevice to another
sendInterApp	Sends an InterApp message from one application to another
sendfxCloud	Sends a message to the fxCloud
sendSMS	Sends an SMS to the user of choice

Caching Framework

[0111] The Caching Framework allows the developers to cache particular contents or files that are accessed frequently in the caching engine. The caching can be done locally in the fxDevice, clustered cache (shared amongst a few fxDevice), or cloud caching. Table 5 highlights some examples of methods/functions that may be available in the Caching Framework.

Table 5: Caching Framework	
Method	Description
storeForward	Enables Store and forward model for a target class of traffic where the target traffic is matched and stored upon a condition and forwarded the condition is relieved
cacheWebSite	Caches the most visited web sites
cacheVideo	Caches the most viewed videos
cacheCloud	Caches the requested content or traffic type in the cloud storage in fxCloud

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0224

Management Framework

[0112] The Management Framework enables the developers to manage the BS platforms and perform administrative procedures. Table 6 highlights some examples of methods/functions that may be available in the Management Framework.

Table 6: Management Framework

Method	Description
upgradeFirmware	Started and executes the firmware upgrade procedure on the fxDevice
upgradeOS	Started and executes the operating system upgrade procedure on the fxDevice
remoteBootup	Performs remote boot up process where the fxDevice works closely with the fxCloud to perform the bootup
remotePower	Allows fxCloudApps to remotely turn on and off a fxDevice or elements in the fxDevice (e.g. the wireless)
gatherStatistics	Instructs fxDevice to gather specific statistics. Various commands may be consolidated for gathering statistics. Number of active users Stats of time spent by users in the cell Stats of quality perceived by the users Neighboring cell info Overall traffic passed through the system Total number of connected devices per frequency, per carrier, per site, per location Handover failure statistics Cloud aggregated data processing and full report Total traffic passed Applications used (meta-data correlated) Bandwidth consumed User's mobility pattern

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0225

setMonitoring	Allows the developer to get reports and statistics of the fxDevice traffics, errors, and neighboring environment. The reporting criteria could be set up as: 1) Period, 2) Event based (reports only if a threshold is passed; a threshold could be based on absolute numbers or deltas), 3) One-time (i.e. the information is pulled once). The exact parameters to get report on may use the same data structures used by the method "getStatistics".

getSystemInfo	Retrieves static and dynamic system info such as: • Hardware info: vendor, model number, capability profile. GPS, Cellular, WAN interfaces, RFID/NFC • Heat temperature • Vendor Specific Information • Software info: OS version, firmware information • CPU load, per-application load (CPU, memory, accelerators) • Loaded apps and status • Memory utilization • Storage utilization • Interface status & utilization (RF, backhaul, management...)
getfxDeviceLocation	Retrieves location of the fxDevice • Location would be presented in GPS coordinates and civil address • Location enhancement: Cloud, map data, and neighbor discovery could help more accurate positioning
getPowerConsumption Rate	Retrieves current power consumption rate of the fxDevice
configureWAN	Configures the WAN settings
retrieveNeighborInfo	Retrieves neighbor information from the fxCloud
getRFIDInfo	Returns the RFID/NFC information of the fxDevice
setRFIDInfo	Sets the RFID/NFC information of the fxDevice

Extensible API Framework

[0113] This framework allows the developer to create custom APIs and make it accessible for other applications (in the fxDevice or in the fxCloud). In turn, the fxCloud could present these APIs using e.g. REST technologies (via the fxCloud

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0226

Northbound Interface) to the developer outside the dSND system. There might be limits put on the APIs exposed through the cloud to avoid potential misuse or security threats. The dRS enforces the security policies defined by the admin (via fxManager). The applications take a role of Client Application (CA) or Server Application Role (SA). The CA and SA could be distributed in the fxDevice or fxCloud. The CA makes requests and SA serves the requests.

[0114] The Extensible APIs could be categorized (as examples) into:

Read-Only Data: this type of APIs allows a CA to read data from SAs without the ability to change or request any particular action from the SAs

Read-Write Data: this type of APIs allows CA application to read data from SAs with the ability to change some of their data but no option for requesting any particular action from SAs;

Procedural: this type of APIs allows CAs to execute a particular procedure in SAs. The rights to change any data still depends on whether the API allows Read-Only Data or Read-Write Data types.

Cloud Management Framework

[0115] The Cloud Management Framework presents a collection of methods enables management of the cloud services and resource. Table 7 highlights some examples of methods/functions that may be available in the Cloud Management Framework.

Table 7: Cloud Management Framework

Method	Description
detectCloudCongestion	Detects congestions on a particular VM
deleteVM	Deletes an active VM permanently
suspendVM	Suspends an active VM but doesn't remove it
activateVM	Activates an already created VM
createVM	Creates a new VM but doesn't activate it

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0227

EXAMPLES OF DAPPS

[0116] The dSDN create an end-to-end programming platform and the possibilities of vApps are only limited by the developers' imagination. An example list of potential vApps are presented in Table 8 below.

Table 8 highlights examples of use cases possible by the Distributed Software Defined Network (dSDN)

Use	Description	Framework {Methods} used
1	Distributed SON: For example, the WiFi AP could search for the least crowded channel to minimize the interference	Wireless {discoverNeighbor, retrieveNeighborInfo, measure nterference, setWirelessConfig, loadBalance}, General Framework, Messaging {sendIntraApp}
2	BS Hot Firmware Upgrade without impact to the basic functions of the BS. For this function to support full hot upgrade without operational impact, the Communication SoC should be able to	Management {upgradeFirmware}
3	BS Hot infOS Upgrade without impact to the basic functions of the BS. For this function to support full hot upgrade without operational impact, the full platform should have dual processor.	Management {upgradeOS}
4	Secure network---based booting. This allows to keep the code securely in the cloud to avoid tampering by the ODM/CM or Man---in---the---Middle attacks	Management {remoteBoot}
5	Turning on or off APs if nobody is in the office. Sensors in the building send information to fxCloudApp which in turn orders power control of the APs	Management {remotePower}
6	Store and forward model for low priority M2M applications where the target traffic is matched and stored in the case of network congestion and forwarded once the network congestion is relieved.	DataPath {matchTraffic, detectCongestion}, Cache {storeForward}
7	Replacing MAC Scheduler	Wireless {defineMacScheduler}
8	Replacing Receiver Algorithms	Wireless {definePHYReceiverAlgorithm}
9	Caching most visited website or video clip in the enterprise	DataPath {identifyTopURL}, Cache {cacheWebSite}
10	The most viewed videos and webcasts are cached in the wireless edge	DataPath {identifyTopVideos}, Cache {cacheVideo}
11	Local Routing: data and voice sessions between two parties on the same cell are routed within the same cell bypassing the core network	DataPath {detectLocalConversation, RouteLocal}
12	Local Routing: data and voice sessions between two parties on the nearby cells are routed locally bypassing the core network	DataPath {detectConversationDetect, routeLocal}
13	Temporary redirection of a data flow for monitoring and security reasons. In this use case, a particular user's traffic is tracked by redirection to a monitoring station. If user is mobile, such profile of tracking and associated redirection is transferred from fxDevice to fxDevice either directly or via the fxCloud	DataPath {detectDevice, matchTraffic, redirectTraffic}
14	Power Calendar: On---demand coverage (triggered by passing traffic thresholds in certain cells or by calendaring): Operators need to design their network for peak usage rates at high costs, leaving their network underutilized most of the time	DataPath {detectCongestion}, Management {remotePower}
15	DOS attack recognition and action could be blocking the user, the app, and/or notifying the user via an SMS message	Security {setDOSDetect, configureFirewall}, Messaging {sendfxCloud}
16	Applying certain ACLs to the BS. For example, the students in the class can only access Facebook between the breaks and access to Facebook is blocked during the class	Security {configureFirewall}
17	Forced offloading of some cellular users (users that have attached to the same cell for a while) to WiFi by changing "something" in the cellular connection to force WiFi connection (Heterogeneous)	Wireless {identifyConnectedDevice, forceOffload}
18	Service continuity and anchoring at eNB for WiFi using the same LTE IP address (Heterogeneous)	DataPath {identifyConnectedDevices, assignAddress}
19	Graphic/general processing for special use cases such as City BS	General Frameworks
20	Application filtering/throttling (secondary conditions: time---based, location---based, subscriber---type, none)	DataPath {matchTraffic}, Security {configureFirewall}
21	Local Services: The enterprise user can discover Bonjour or UPnP devices and services (e.g. printer) even when connected to LTE/3G small---cell in the enterprise. In this case, the small cell or fxDevice pro---actively listens to Bonjour and UPnP discovery and advertisement messages (SSDP) and cache the available services until a device request for such information. In the case of Bonjour, fxDevice could listen to Multicast DNS (mDNS) messages sent over cellular connection (LTE/3G) and multicast them on the LAN or WLAN interfaces.	DataPath {matchTraffic, routeLocal}
22	Local Services: The premise---owner (e.g. airport, hotel) can provide ads under the browser for sponsored WiFi or cellular (3G/LTE) access over multi---mode small cell. The premise owner could use this service to advertise special offers or third party ads. This technique uses HTML <script> for example to add special content to the bottom of the page.	DataPath {matchTraffic, addScript}

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0228

23	Distributed Content Distribution Network (CDN). In this use case, the fxDevices function as CDN that bring the contact closer to the end users.	DataPath {matchTraffic}, Cache {cacheVideo}
24	In case of national security, the service provider may limit the access to the network so only the law enforcement can use the cellular network. In this case, the admin defines a Target Area (TA) using an fxCloudApp in the fxCloud where the emergency lockdown needs to be applied. The IMSI/MSISDN number of law enforcement mobile devices are sent to the fxDevice in the TA which in turn enforce the policy. The policy may allow the public to just sent SMS while the law enforcement could have full access and priority to the system.	DataPath {matchTraffic, setPriority}, Security {configureFirewall}
25	Emails and web browsing monitored by the cloud services assisted by the agent in the dSDN network. In this case, the fxDevice detects certain traffic (email or web in this case) and redirects them to a cloud service for security cleansing.	DataPath {matchTraffic, redirectTraffic}
26	Analytics pre---analysis: The data collected from M2M devices could be processed (noise could be filtered out) and packaged (compressed) for consumption by cloud services	DataPath {matchTraffic}, General Framework, Messaging {sendfxCloud}
27	Certain traffic could be marked with QoS DSCP codes for further processing in the network	DataPath {matchTraffic, markTraffic}
28	Proprietary SP services and applications that run purely between the phone and the BS . For example, the SP may implement a certain application for the law enforcement to get direct feed of surveillance cameras in a target area (TA).	DataPath {matchTraffic}, Security {configureFirewall}, General Framework
29	IP---PBX implementation in the small---cell or WiFi AP for enterprise use. This would allow a simple out of a box solution. The VoIP phones connect to the fxDevice via the LAN connections. The WAN connection could connect to an fxCloudApp for PSTN calls	DataPath {matchTraffic}, General Framework
30	Network management system can add its own agent/probe to collect and consolidate data and transmit it to its cloud element using any protocol	Management {gatherStatistics, getSystemInfo, control}, General Framework
31	Application based performance monitoring on the wireless link (number of packets sent in idle mode, retransmit rate…). Such information could be shared with the app developer to optimize its application	Management {gatherStatistics}, General Framework
32	Seamless network sharing and mergers. This is the case where two Network Operators would like to share or merge their network. Using dSDN's infManager, the network administrators could merge the network administration control. This process may include setting up the desirable default fxApp packages of the network operator	fxManager Portal Service
33	Following a user to understand where it faces coverage issues or call drops. Once discovered, the system could send a message to the user acknowledging an issue and that the SP is trying to fix this. The information is uploaded to the cloud for further analysis	Management {gatherStatistics}, General Framework, Messaging {sendSMS}
34	Reporting: This allows the developer to get reports and statistics of the fxDevice traffics, errors, and neighboring environment. The reporting criteria could be set up as: 1) Period, 2) Event based (reports only if a threshold is passed; a threshold could be based on absolute numbers or deltas), 3) One---time (i.e. the information is	Management {setMonitoring}, General Framework
35	Proprietary security protocol between the application in the mobile device and the BS in enterprise deployment. This would allow SP to create enterprise---class LTE for indoor using small cells	Security {encryptContent, authenticateUser, Integrity, configureFirewall}, DataPath {RouteLocal}, General Framework
36	The femto cell at home that can't scan neighboring cells, gets a list of neighboring cells from the fxCloud or neighbors to broadcast to improve outbound handoffs	Wireless {discoverNeighbors, retrieveNeighborInfo, setNeighborCells}, Management {retrieveNeighborInfo}
37	IMS---capable femto enabling various SP services to initiate or transferred between home network and mobile devices	Messaging {sendfxCloud}, General Framework
38	Integrated smart home with multi---mode femto/WiFi router. For example, the application in the smart home router (fxDevice) allows for coordination of home physical security service (such as ADT services). With the fxCloudApp, the user could control the security status of his/her home and remotely control the setting of the	Wireless {homeAreaNetworking}, Messaging {fxCloud}, General Framework
39	Home CCTV DVR implementation in an integrated home router. The camera feeds are recorded on the same router caching engine	Cache {cacheVideo}, Messaging {sendfxCloud}, General Framework
40	Enterprise CCTV DVR implementation in an integrated router with cloud backup. The camera feeds are recorded in the cloud.	Cache {CacheVideo, CacheCloud}, Messaging {fxCloud}, General Framework
41	User traffic separation into various MPLS/LSP (integrating VPN and BS). In this case, users are classified into groups and each group is tunneled to the core network. For example, law enforcement use the encrypted tunnel while the other users use the default tunnel	DataPath {Match, SetupTunnel, RouteTunnel}
42	Dynamic spectrum allocation where a BS borrows unused spectrum from the neighboring BS in case of temporary congestions	DataPath {detectCongestion}, Wireless {discoverNeighbors, setWirelessConfig}, Management {retrieveNeighborInfo, borrowCarrier}, Messaging {fxCloud}

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0229

43	Proxy ANDSF (Access Network Discovery and Control Function) in the BS. When the UE tries to access the ANDSF, the Proxy ANDSF fxDeviceApp in the BS intercept the ANDSF messages and replies on behalf of the ANDSF. This would speed up the WiFi network discovery for example.	DataPath {matchTraffic}, General Framework, Messaging {sendfxCloud}
44	HD Audio calling: audio codecs in BS allows the phones with the right capabilities to communicate with HD Audio quality calling. If one of the mobile devices can't handle the HD quality audio, the BS performs the transcoding of the audio.	DataPath {matchTraffic}, General Framework
45	Enabling mobility on White Space Spectrum: The latest white space database will be downloaded. The mobile devices with special software applications make requests to the BS to retrieve the local available white space frequencies. This would allow the mobile device to handoff seamless between BS using the white space spectrum. The BS would advertise such a capability to the UEs. The secure communication allows peer---to---peer communication between the UE and the BS for certain applications such as White Space Channel lookup or site acquisition tools for field engineering and RF planning.	Messaging {sendfxCloud}, General Framework
46	Site acquisition tool where the technician uses a software tool on his mobile phone to interact with BSs to determine the best place to deploy the new BS. The phone app interests with fxDeviceApps in BSs and fxCloudApp. For example, fxDeviceApp performs air interface quality and gives a priority to traffic generated by for this	Messaging {sendfxCloud}, General Framework. DataPath {matchTraffic, setPriority}
47	DNS caching at BS to accelerate DNS lookup for subsequent users. This would improve the user experience by reducing response delays	DataPath {matchTraffic}, General Framework
48	Continuous SLA monitory for certain protocols, applications or general link quality for jitter, delays, bandwidth	DataPath {matchTraffic}, Management {gatherStatistics}, General Framework
49	Enterprise users can use the small cell deployed in their enterprise free of charge. The DPI engine in fxDevice classifies the traffic and identifies the enterprise private data and they would be routed locally and charging records are generated by the BS (the SP decides how to bill the enterprise user and it may decide to make such usage free	DataPath {matchTraffic, routeLocal}
50	On---premise small---cell aggregation where on fxDevice takes the responsibility of aggregating multiple small---cells on the premise and presenting them as a single BS/Node---B/eNodeB to the core network. In this case the aggregating fxDevice (AfxE) may take responsibility of local radio resource manager (RRM) or self---optimizing network (SON) server. AfxE can also implement and enforce policies concerning local service access on the premise (enterprise) as well as enable seamless handover between cellular technologies and on---premise (enterprise) local area network (LAN) for example using WiFi technology.	General Frameworks, Wireless {discoverNeighbors, retrieveNeighborInfo, setNeighborCells, setWirelessConfig}, Messaging {sendIntraApp}
51	Self adjusting fxCloud compute resources: The fxCloud creates and tears down virtual machines (VMs) based on the traffic load measured by itself or the packet processing and data plane engine. It is also possible that the VM adjustments are made based on the rush hour or pre---configured hours of the day or manually by the administrator.	CloudManagement {removeVM, createVM, detectCloudCongestion, actiavteVM, suspendVM}
52	Core Network Function Virtualization (cNFV) using distributed Software Defined Networking (dSDN): In the case, once an fxDevice is commissioned, its fxDeviceApp0 (i.e. firmware) uses predefined DNS names to discover the rest of the core network (e.g. MME, S---GW, etc.). The core network elements could be created and	General Frameworks, Messaging {sendInterApp, sendIntraApp), CloudManagement {deleteVM, suspendVM, createVM, activateVM, detectCloudCongestion}
53	Smart display and digital signage as wireless base stations: this use case allows combining the capabilities of a base station and the smart display in shopping malls, airports, and enterprises. There is a already electric power and connectivity for the smart display that the base station function could use.	General Frameworks
54	Tracking a mobile user at base station level even if the UE is in the idle mode. Currently in the idle mode, the MME, SGSN, or VLR/MSC can only know the location of the UE in with a accuracy of LA/RA/URA which is a very large area. The administrator uses the fxCloudApp and requests tracking of the user/UE (the admin may provide target area where the user/UE may be located; e.g. zip code or town name). The fxCloudApp contacts its fxDeviceApp in the fxDevices in the target area (TA) requesting information of a particular IMSI/MSISD/TMSI. Once an fxDevice responds indicating knowledge of the user location, the fxCloudApp directs that serving fxDevice to inform the fxDeviceApp in the next fxDevice of the instructions to track the user and contact fxCloudApp when they user hands off to due to its mobility. fxCloud could create user friendly interfaces and presentations to show for example the direction the user is moving, predict next location of the user, and estimate arrival of the user to a certain location	DataPath {matchTraffic}, Messaging {sendIntraApp, sendfxCloud}
55	In case of emergency, fxDevices in a target area block all public traffic and only allow law enforcement sessions using specially encryption and authentication between the UE and the fxDevice or fxCloud (depending on the requirements)	DataPath {matchTraffic, setPriority}, Security {configureFirewall, encryptTraffic, authenticateUser}, General Framework

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0230

56	SON triggered by the data traffic consumption. In this case, the mobile cell configuration is adjusted based on the current traffic of the cell. It could for example force handoff some users (high users or users moving fast) to neighboring cells.	DataPath {identifyTopDevices}, Wireless {forceHandoff}
57	Time based QoS and policy. In this use case, the infManager pushes certain policies in target area (TA). Such polices could be turning on extra capacity (frequency channels) or QoS policies and traffic prioritization on the WAN like	DataPath {matchTraffic, setPriority}, Wireless {discoverNeighbors, setWirelessConfig}
58	Fast moving users detection: in this use case, the application in the BS measures the speed of the user (based on the time was handed off to this cell until it was handed over to the next cell). This information could be used by the fxDevice or fxCloud to force handoff the mobile user to a larger cell to reduce handoff rates	General Frameworks, Wireless {identifyTopMovers, forceHandoff}
59	Emergency broadcast of information and live camera feeds. In case of emergency, the enterprise network administrator temporarily block all traffic and only allow emergency related traffic across the target area in the enterprise network. For example, the network admin broadcasts the camera feeds and alerts over the WiFi access points using multicast IP techniques. The client devices and digital sinages can listen to that particular multicast and show the camera feeds and emergency alerts	DataPath {matchTraffic, setPriority}, Security {configureFirewall}, General Framework
60	Dynamic advertising in shopping malls. The shopper would use an app from the shopping mall. When the shopper brings up or refreshes the app, the dSDN allows the shopping mall to pinpoint the exact indoor/outdoor location of the user (e.g. the shop the shopper is in) and send special coupon and ads for that store to the shopper.	General Frameworks, Wireless {positionDevice}, Messaging {sendIntraApp, sendfxCloud}, Security {configureFirewall}, DataPath {matchTraffic, redirectTraffic}
61	Smart tour guide. The tourist would use an app from the tourism board or museum. When the user brings up or refreshes the app, the dSDN allows to pinpoint the exact location of the tourist (e.g. the painting he/she is looking at) and send tour guide information (e.g. audio, video, hypertext). dSDN allows to localize the positioning and	General Frameworks, Wireless {positionDevice}, Messaging {sendIntraApp, sendfxCloud}, Security {configureFirewall}, DataPath {matchTraffic, redirectTraffic}
62	General hotspot landing page	DataPath {matchTraffic, redirectTraffic}, Security {authenticateUser}
63	Proprietary wireless mesh implementation	Wireless {discoverNeighbor, retrieveNeighborInfo, measureInterference, setWirelessConfig}, General Framework, Messaging {sendIntraApp}
64	Power and energy consumption map of the network. A real---time fxCloudApp pulls the power consumption details from each fxDevice periodicaly or on demand. The pulled data then are combined with the location information and correlated into a map to present a viewable consumption map	Management {getfxDeviceLocation, getPowerConsumptionRate}
65	Network deployment optimization. This application allows network operator to get advice for best optimal reconfiguration and/or redeployment of the network. For example, the network admin could get suggestion that based on the power consumptions and coverage perceived and spectrum utilization, it might be better to replace a particular group of smaller cells with a single macro---cell. This app could also pull information from larger OEMs based on the subject deployment scenario (the OEMs either provide this information statically to the tool or provide network APIs for use by this app e.g. using REST protocol)	Management {getfxDeviceLocation, getPowerConsumptionRate, gatherStatistics}
66	High availability. This app allows two or more fxDevice to form a redundancy cluster where they participate in statueful redundancy and load---balancing	Messaging {sendIntraApp}, General Framework
67	Bandwidth Bursting. This feature enables the dynamic adjustment of backhaul or WAN link bandwidth as the demands change in the enterprise or public wireless deployments. In this case, the fxDeviceApp in the fxDevice monitors the WAN link utilization, once a preset threshold percentage has reached, it messages its sister fxCloudApp which in turn would send a request to the fxDeviceApp to increase or decrease the WAN bandwidth. If additional backend reconfiguration is needed, the fxCloudApp performs those additional changes prior to informing the fxDeviceApp of the bandwidth	Management {getStatistics, configureWAN}
68	Wireless link quality measurement by the small---cell or WiFi AP: The wireless SoC in the small---cell or WiFi AP continuously monitors the link quality for power adjustment and handover reasons. Such information could be exposed to the app developers to create cell---level analytics and reports without the need to have test--- drives or field testing of service quality	Management {gatherStatistics}, General Framework
69	Data Path Consolidation: An application on the fxDevice discovers identical data feeds from a cloud server to two or more users. The fxDeviceApp in the fxDevice can discover this potential duplicate flows at the session establishment (via a proxy function) and it would consolidate those sessions into one on the uplink towards the cloud server. A good example of this use case is the web---based desktop sharing where different many users in the same office site get the same feed from the cloud server.	FastPath {detectIdenticalFlows, aggregateFlows}

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0231

[0117] The following references are herein specifically incorporated by reference in their entirety into this disclosure:

[1] 3GPP TS 23.002, “Network Architecture”;
[2] 3GPP TS 23.402, “Architecture enhancements for non-3GPP accesses”;
[3] 3GPP TS 23.401, ~~“General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access”;~~
, “General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access”;
[4] Open Base Station Architecture Initiative (OBSAI); and
[5] SPDY Protocol (http://datatracker.ietf.org/doc/draft-mbelshe-httpbis-spdy/).

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0232

CLAIMS

1. A plurality of network elements in a packet system, said network elements comprising:

a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp;

a flexible cloud platform having at least one fxCloudApp, wherein the at least one fxDeviceApp and at least one fxCloudApp are in secure communication to form a distributed application; and

an application management portal which manages life cycle of the distributed applications.

2. The system of claim 1, wherein each of the flexible network device, flexible cloud platform and application management portal include an access network interface unit configured to send and receive communications and a processor with a memory associated with the network interface unit.

3. The system of claim 1 further comprising:

an infrastructure application market place in communication with the application management portal, said infrastructure market place capable of providing tested and certified applications to the application management portal.

4. The system of claim 1, wherein the at least one fxDeviceApp is responsible for a plurality of functional capabilities of the flexible network device and is capable of at least one of the group consisting of: updating the firmware of the flexible network device; updating default communication protocols of the flexible network device; initializing flexible network device settings; boot loading the flexible network device; providing

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0233

security digital certificates; providing OS and kernal upgrades; and upgrading and commissing agents for the fxDeviceApp upgrades and related procedures.

5. The system of claim 1, wherein the at least one fxCloudApp is responsible for a plurality of functional capabilities of the flexible cloud platform and is capable of at least one of the group consisting of: upgrading and commissioning agents for the fxCloudApp updates and related procedures; hypervisor upgrades; virtual machine and/or isolated execution environment lifecycle management; security digital certificate management; OS and kernel upgrades; and default communication protocols.

6. The system of claim 1, wherein the application management portal is capable of at least one of the group consisting of: receiving new applications from the infrastructure application market place; test said distributed application prior to deployment; provision said distributed application; and deprovision said distributed application.

7. The system of claim 1, wherein the flexible network device is an LTE eNB and the flexible cloud platform hosts an LTE core network functions in separate virtual machines from the group consisting of: MME, S-GW, and P-GW.

8. The system of claim 1, wherein the flexible network device and flexible cloud platform further comprise:
a virtual fabric which provides a secure communication layer for said at least one fxDeviceApp and said at least one fxCloudApp.

9. The system of claim 1, further comprising:
a second flexible network device which includes at least one fxDeviceApp which is the same as the at least one fxDeviceApp in the first flexible network device and where each of the fxDeviceApps can communicate directly.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0234

10. The system of claim 1, wherein the flexible network device and flexible cloud platform further comprise:

a distributed resource service (DRS) which controls access to a plurality of resources upon receiving instructions from the application management portal.

11. The system of claim 10, wherein the resources include at least one of a group consisting of: firewall; data and statistics; storage; compute; load evaluator; general settings; routing engine; virtual machine master controller; power controller; and wireless engine.

12. The system of claim 1, wherein the flexible network device and flexible cloud platform further comprise:

a distributed resource service which is capable of providing at least one service from a group consisting of: exposing APIs to other applications; configuring and managing platform resources; policy enforcement and authorization of applications access to platform resource and other app's APIs; and policy conflict resolution.

13. The system of claim 1, wherein the flexible network device and flexible cloud platform further comprise:

a distributed content service wherein contents generated by the at least one fxDeviceApp may be shared with the at least one fxCloudApp.

14. The system of claim 1, wherein the flexible network device and flexible cloud platform further comprise:

a distributed notification service wherein notice of a particular event is sent from the at least one fxDeviceApp to the at least one fxCloudApp.

15. The system of claim 10, wherein the DRS further includes a load controller; and wherein the load controller is adapted to monitor loads on the at least one fxDeviceApp and at least one fxCloudApp and effect change in accordance with thresholds received from the application management portal.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0235

16. The system of claim 1, wherein the fxDevice, at least one fxDeviceApp, flexible cloud platform, and the at least one fxCloudApp each have unique identities.

17. The system of claim 1, wherein the at least one fxDeviceApp and and at least one fxCloudApp are given an access level to determine which APIs can be accessed.

18. The system of claim 1, wherein the at least one flexible network device and flexible cloud platform are adapted to verify the authenticity and integrity of updates.

19. A plurality of network elements in a packet system, said network elements comprising:

a flexible cloud platform having a plurality of virtual machines, wherein each of the virtual machines has at least one fxCloudApp; and

a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp, wherein the at least one fxDeviceApp and fxCloudApp are in secure communication to form a distributed application.

20. A plurality of network elements in a packet system, said network elements comprising:

a flexible cloud platform having a plurality of zones, wherein each of the plurality of zones has at least one fxCloudApp; and

a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp, wherein the at least one fxDeviceApp and fxCloudApp are in secure communication to form a distributed application.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0236

ABSTRACT

The Distributed Software Defined Network (dSDN) disclosed herein is an end-to-end architecture that enables secure and flexible programmability across a network with full lifecycle management of services and infrastructure applications (fxDeviceApp). The dSDN also harmonizes application deployment across the network independent of the hardware vendor. As a result, the dSDN simplifies the network deployment lifecycle from concept to design to implementation to decommissioning.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0237

Electronic Acknowledgement Receipt

EFS ID:	42492135
Application Number:	17142983
International Application Number:	
Confirmation Number:	2943
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Customer Number:	112449
Filer:	Christopher James Horgan
Filer Authorized By:	
Attorney Docket Number:	TAAG001CONT3
Receipt Date:	19-APR-2021
Filing Date:	06-JAN-2021
Time Stamp:	17:41:46
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	Applicant Response to Pre-Exam Formalities Notice	TAG001CONT3_RESPONSE_NOTICE.pdf	26512 810d854b07192b065d963414891111617cf94ae7d	no	1

Warnings:

Information:					
2	Specification	TAAG001CONT3_SPECIFICATION_CLEANv2.pdf	2553883 d1a223b6a12c5b8eca381ebfde9683535d0a94ad	no	59
Warnings:					
Information:					
3	Specification	TAAG001CONT3_SPECIFICATION_MARKED_UPv2.pdf	2560383 0561bc09996e0e79ea65239a45241542a1bb0978	no	59
Warnings:					
Information:					
			Total Files Size (in bytes):	5140778	

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.

DISTRIBUTED SOFTWARE DEFINED NETWORKING

CROSS-REFERENCE TO RELATED PATENT APPLICATIONS

[0001] The present invention claims priority from U.S. Provisional Patent Application Serial No. 61/834,807, filed June 13, 2013 the disclosure of which is herein specifically incorporated by reference in its entirety.

FIELD OF THE INVENTION

[0002] This invention relates to a network architecture that facilitates secure and flexible programmability between a user device and across a network with full lifecycle management of services and infrastructure applications.

BACKGROUND

[0003] Existing network solutions are built on proprietary hardware and software. Network Operators & Information Technology (NOIT) can only configure the proprietary network infrastructure provided by vendors and are unable to add new and customized features or capabilities. As a result, the NOIT can only add new desired features by making such requests to their infrastructure vendors or pursuing standardization processes. But these existing approaches are time consuming and resource intensive.

SUMMARY OF THE INVENTION

[0004] An aspect of the disclosure herein is a plurality of network elements in a packet system, said network elements comprising: a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp; a flexible cloud platform having at least one fxCloudApp, wherein the at least one fxDeviceApp and at least one fxCloudApp are in secure communication to form a distributed application; and an application management portal which manages life cycle of the distributed applications.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0240

[0005] Another aspect of the disclosure is a plurality of network elements in a packet system, said network elements comprising: a flexible cloud platform having a plurality of virtual machines, wherein each of the virtual machines has at least one fxCloudApp; and a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp, wherein the at least one fxDeviceApp and fxCloudApp are in secure communication to form a distributed application.

[0006] Another aspect of the disclosure is a plurality of network elements in a packet system, said network elements comprising: a flexible cloud platform having a plurality of zones, wherein each of the plurality of zones has at least one fxCloudApp; and a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp, wherein the at least one fxDeviceApp and fxCloudApp are in secure communication to form a distributed application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Figure 1 demonstrates the network architecture of a 4G system.

[0008] Figure 2 illustrates a network element (or system) upon which the Distributed Software Defined Network operates.

[0009] Figure 3 illustrates a Distributed Software Defined Network in accordance with the present disclosure.

[0010] Figure 4 is an exemplary view of the software layers in the Flexible Operation System (fxOS) Architecture.

[0011] Figure 5 illustrates the FastPath architecture.

[0012] Figure 6 presents a potential software layer of the fxOS co-existing alongside a general purpose OS.

[0013] Figure 7 is an exemplary view of the fxStore Portal.

[0014] Figure 8 demonstrates a shared fxCloud deployment.

[0015] Figure 9 demonstrates a Zoned Deployment.

[0016] Figure 10 demonstrates another deployment option as applied to cellular networks (especially Long Term Evolution (LTE)).

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0241

[0017] Figure 11 shows an example of a signaling flow chart for the fxDevice Commissioning Procedure.

[0018] Figure 12 shows an example procedure for Application Provisioning.

[0019] Figure 13 presents the state machine for the application upgrade process.

[0020] Figures 14A-14C are examples of the Distributed Switch Controller during Upgrade.

[0021] Figure 15 shows how fxVF in the fxCloud and fxDevice.

[0022] Figure 16 demonstrates the logical interfaces between the platform resources and applications.

[0023] Figures 17A and 17B demonstrate example procedures for cloud breathing procedures.

DETAILED DESCRIPTION

[0024] The Distributed Software Defined Network (dSDN) disclosed herein is an architecture that enables secure and flexible programmability across a network with full lifecycle management of services and infrastructure applications (fxDeviceApp). The dSDN also harmonizes application deployment across the network independent of the hardware vendor. As a result, the dSDN simplifies the network deployment lifecycle from concept to design to implementation to decommissioning.

DEFINITIONS, ACRONYMS & ABBREVIATIONS

[0025] The following terms, acronyms, abbreviations and descriptions are explained below and are used throughout the detailed description of the dSDN:

TERM	DESCRIPTION
2G	2nd Generation Cellular Technology
3G	3Rd Generation Cellular Technology
4G	4th Generation Cellular Technology
AP	WiFi Access Point
API	Application Programming Interface

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0242

AR	Application Repository
ASIC	Application Specification Integrated Circuit
AuC	Authentication Center
BS	Base Station
BSC	Base Station Controller
BTS	Base Station
CDN	Content Distribution Network
CN	Core Network
CNIE	Core Network Internet Edge
CPU	Central Processing Unit
dAP	Distributed Application Package
dApp	Distributed App
dCP	Distributed Content Service
dNS	Distributed Notification Service
DNS	Domain Name System
DPDK	Data Plane Development Kit
DPI	Deep Packet Inspection
dRS	Distributed Resource Service
DSCP	Differentiated Service Code Point (QoS)
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
EIR	Equipment Identity Register
eNB	Enhanced Node B (4G)
FIB	Flow Information Base
FQDN	Fully Qualified Domain Name
fxApp	FleXible Application, which has a fxDeviceApp and fxCloudApp component. fxDeviceApp and fxCloudApp of one application represent fxApp.
fxDeviceApp	FleXible Application which runs on fxDevice
fxBS	FleXible Base Station that is a BS that is powered by fxOS (an example of an fxDevice)

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0243

fxCloud	FleXible Cloud that interacts with fxDevice and on other network elements on the northbound interface
fxCloudApp	FleXible Cloud Application that is the application that runs on fxCloud and can be associated to one or more fxDeviceApp in fxDevice
fxManager	FleXible Manager that manager the fxOS lifecycle (provisioning, usage, de-provisioning)
fxOS	FleXible Operation System which run on the fxDevice as the OS and firmware
fxSDK:	FleXible Software Development Kit
fxSimulator	FleXible Simulator
fxStore	FleXible Store that presents fxApp to the network
administratorsGERAN	GPRS Edge Radio Access Network
GGSN	Gateway GPRS Support Node (used in core network of 2G/3G systems)
GMSC	Gateway Mobile Switching Center
GPRS	General Packet Radio Service
GPS	Global Positioning System
GTP	GPRS Tunneling Protocol
GW	Gateway
HAL	Hardware Abstraction Layer
HD	High Definition
HLR	Home Location Register
HSS	Home Subscriber Server
IAC	Inter Application Communication

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0244

IMS	IP Multimedia Sub-system
IMSI	International Mobile Subscriber Identifier
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
JVM	Java Virtual Machine
LA	Location Area
LAN	Local Area Network
LSP	Label Switching Protocol
M2M	Machine to Machine
MAC	Medium Access Control
MCP	Multi Chip Packaging
MIMO	Multi Input Multi Output
MME	Mobility Management Entity (used in core network of LTE)
MNO	Mobile Operator Network
MPLS	Multi Protocol Label Switching
MSC	Mobile Switching Center
MSISDN	Mobile Station
NAS	Non-Access Stratum
NB	Node B (3G)
NFC	Near Field Communication
NFV	Network Function Virtualization
NOIT	Network Operations & Information Technology; this term network broadly includes carriers, service providers, enterprise networks, and designated/contacted 3rd party administrators.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0245

OS	Operating System
OTT	Over the Top
P-GW	Packet Gateway
PaaS	Platform-as-a-Service
PBX	Private Branch exchange (telephony)
PDG	Packet Data Gateway
PDN	Packet Data Network
PHY	Physical Layer
PLMN	Public Land Mobile Network
PnP	Plug and Play
PP	Packet Processor
PR	Platform Resources
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RA	Routing Area
RAN	Radio Access Network of cellular networks
RAPI	Resources API
RFID	Radio Frequency Identifier
RNC	Radio Network Controller
S-GW	Serving Gateway
S1-AP	S1 Interface Application Protocol (an LTE protocol between MME and eNodeB)
SaaS	Software-as-a-Software
SDK	Software Development Kit
SDN	Software Defined Network
SGSN	Serving GPRS Support Node (used in core

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0246

	network of 2G/3G systems)
SIM	Subscriber Identity Module
SiP	System in a Package
SMS	Short Message Service
SoC	System on a Chip
SON	Self-Optimizing Network
SSDP	Simple Service Discovery Protocol
TA	Target Area
TNE	Test Network Environment
TTM	Time to Market
UMTS	Universal Mobile Telecommunication System
UPnP	Universal Plug and Play
URA	UMTS Routing Area
URI	Uniform Resource Identifier
UTRAN	UMTS Terrestrial Radio Access Network
VLR	Visitor Location Register
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WLAN	Wireless Local Area Network
XMPP	Extensible Messaging and Presence Protocol

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0247

[0026] Database. One or more large structured sets of persistent data maintained upon a computer system organized and structured according to a software system defining rules for organization as well as responding to queries to read, write, or modify data as well as provide statistical information regarding the contained data. As used in this disclosure in describing the dSDN, a database may be either a single unified system or a distributed system wherein certain database elements are located upon different systems or servers which may be in different physical locations, acting in harmony to appear as one unified database. Where databases are described, it will be understood by one of ordinary skill in the art that (i) alternative database structures to those described may be readily employed, and (ii) other memory structures besides databases may be readily employed. Any illustrations or descriptions of any sample databases presented herein are illustrative arrangements for stored representations of information. The database formats may include relational databases, object-based models and/or distributed databases which could be used to store and manipulate the data types described herein. Likewise, object methods or behaviors of a database can be used to implement various processes, such as the described herein. In addition, the databases may, in a known manner, be stored locally or remotely from a device which accesses data in such a database.

[0027] The term "determining" and grammatical variants thereof (e.g., to determine a price, determining a value, determine an object which meets a certain criterion) is used in an extremely broad sense. The term "determining" encompasses a wide variety of actions and therefore "determining" can include calculating, computing, processing, deriving, investigating, looking up (e.g., looking up in a table, a database or another data structure), ascertaining and the like. Also, "determining" can include receiving (e.g., receiving information), accessing (e.g., accessing data in a memory) and the like. Also, "determining" can include resolving, selecting, choosing, establishing, and the like. The term "determining" does not imply certainty or absolute precision, and therefore "determining" can include estimating, extrapolating, predicting, guessing and the like. The term "determining" does not imply that mathematical processing must be performed, and does not imply that numerical methods must be used, and does not imply that an algorithm or process is used.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0248

[0028] The functionality and/or the features of a single device that is described may be alternatively embodied by one or more other devices which are described but are not explicitly described as having such functionality/features. Thus, other embodiments need not include the described device itself, but rather can include the one or more other devices which would, in those other embodiments, have such functionality/features.

[0029] Devices that are described as in "communication" with each other or "coupled" to each other need not be in continuous communication with each other or in direct physical contact, unless expressly specified otherwise. On the contrary, such devices need only transmit to each other as necessary or desirable, and may actually refrain from exchanging data most of the time. For example, a machine in communication with or coupled with another machine via the Internet may not transmit data to the other machine for long period of time (e.g. weeks at a time). In addition, devices that are in communication with or coupled with each other may communicate directly or indirectly through one or more intermediaries.

[0030] Although process (or method) steps may be described or claimed in a particular sequential order, such processes may be configured to work in different orders. Further, some steps may be performed simultaneously despite being described or implied as occurring non-simultaneously (e.g., because one step is described after the other step) unless specifically indicated. Moreover, the illustration of a process by its depiction in a drawing does not imply that the illustrated process is exclusive of other variations and modifications thereto, does not imply that the illustrated process or any of its steps are necessary to the embodiment(s), and does not imply that the illustrated process is preferred. Where a process is described in an embodiment the process may operate without any user intervention.

[0031] It will be readily apparent to one of ordinary skill in the art that the various processes of the dSDN described herein may be implemented by, e.g., appropriately programmed general purpose computer(s), special purpose computer(s) and computing device(s). Typically a processor (e.g., one or more microprocessors, one or more microcontrollers, one or more digital signal processors) will receive instructions (e.g., from a memory or like device), and execute those instructions,

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0249

thereby performing one or more processes defined by those instructions. Instructions may be embodied in, e.g., one or more computer programs, one or more scripts.

[0032] A "processor" means one or more microprocessors, central processing units (CPUs), computing devices, controllers, microcontrollers, digital signal processors, or like devices or any combination thereof, regardless of the architecture (e.g., chip-level multiprocessing/multi-core, Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC), Microprocessor without Interlocked Pipeline Stages, pipelining configuration, or simultaneous multithreading).

[0033] Further, programs that implement methods described herein may be stored and transmitted using a variety of media (e.g., computer readable media) in a number of manners. In some embodiments, hard-wired circuitry or custom hardware may be used in place of, or in combination with, some or all of the software instructions that can implement the processes of various embodiments. Thus, various combinations of hardware and software may be used instead of software only to implement the embodiments.

[0034] The term "non-transitory computer readable medium" in this disclosure refers to any medium, a plurality of the same, or a combination of different media, that participate in providing data (e.g., instructions, data structures) which may be read by a computer, a processor or a like device. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media may include, for example, optical or magnetic disks and other persistent memory. Volatile media may include dynamic random access memory (DRAM), which typically constitutes the main memory. Transmission media may include coaxial cables, copper wire and fiber optics, including the wires that comprise a system bus coupled to the processor. Transmission media may include or convey acoustic waves, light waves and electromagnetic emissions, such as those generated during radio frequency (RF) and infrared (IR) data communications. Common forms of non-transitory computer-readable media in which the dSDN may be implemented include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, any other magnetic medium, a CD-ROM, DVD, any other optical medium, punch cards, paper

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0250

tape, any other physical medium with patterns of holes, a Random Access Memory (RAM), a programmable read only memory (PROM), an erasable programmable read only memory (EPROM), a flash electrically erasable programmable read only memory (FLASH-EEPROM), any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0035] Various forms of computer readable media in this disclosure may be involved in carrying data (e.g. sequences of instructions) to a processor. For example, data may be (i) delivered from RAM to a processor; (ii) carried over a wireless transmission medium; (iii) formatted and/or transmitted according to numerous formats, standards or protocols, such as Ethernet (or IEEE 802.3), Bluetooth, and Transmission Control Protocol/Internet Protocol ("TCP/IP"), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), and 2G/3G; and/or (iv) encrypted to ensure privacy or prevent fraud in any of a variety of ways well known in the art.

[0036] Thus in some embodiments in this disclosure a description of a process may be a description of a non-transitory computer-readable medium storing a program for performing the process. The computer readable medium may store (in any appropriate format) those program elements which are appropriate to perform the method.

[0037] In an embodiment, a server computer, network element or centralized authority may not be necessary or desirable. For example, an embodiment may be practiced on one or more devices without a central authority. In such an embodiment, any functions described herein as performed by the server computer or data described as stored on the server computer may instead be performed by or stored on one or more such devices.

[0038] In interpreting the present application (which includes the claims), one of ordinary skill in the art shall refer to the prosecution history of the present application, but not to the prosecution history of any other patent or patent application, regardless of whether there are other patent applications that are considered related to the present application, and regardless of whether there are other patent applications that share a claim of priority with the present application.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0251

[0039] Numerous embodiments are described in the present application, and are presented for illustrative purposes only. The described embodiments are not, and are not intended to be, limiting in any sense. One of ordinary skill in the art will recognize that the disclosed embodiment(s) may be practiced with various modifications and alterations, such as structural, logical, software, and electrical modifications. Although particular features of the disclosed invention(s) may be described with reference to one or more particular embodiments and/or drawings, it should be understood that such features are not limited to usage in the one or more particular embodiments or drawings with reference to which they are described, unless expressly specified otherwise.

[0040] The present disclosure is not a literal description of all embodiments of the invention(s). Also, the present disclosure is not a listing of features of the invention(s) which must be present in all embodiments. A description of an embodiment with several components or features does not imply that all or even any of such components/features are required. On the contrary, a variety of optional components are described to illustrate the wide variety of possible embodiments of the present invention(s).

[0041] The following description has been presented for purposes of illustration and description and is not intended to be exhaustive or to limit the embodiments to the precise form disclosed. Many modifications and variations are possible in light of the teachings disclosed herein.

[0042] The embodiments were chosen and described to explain principles of operation and their practical applications. However, the scope of the invention is to be defined by the claims.

[0043] The major technical reason for inflexibility in network infrastructure has been related mostly to the proprietary hardware and rigid software architecture in the existing network elements (routers, switches, gateways, cellular base-stations, WiFi access points, etc.). With recent advancements in semiconductor device manufacturing, there is no longer a cost barrier to add computing power to many electronics in the market including networking infrastructure. The addition of such compute capabilities would open up new opportunities for programmable platforms in the network and ultimately creating more flexible network architecture

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0252

and business models.

[0044] The Distributed Software Defined Network (dSDN) disclosed herein is an end-to-end architecture that enables secure and flexible programmability across a network with full lifecycle management of services and applications (fxApp). The dSDN also harmonizes fxApp deployment across the network independent of the hardware vendor. As a result, the dSDN simplifies the network deployment lifecycle from concept to design to implementation to decommissioning. In this disclosure, the dSDN is applied to the wireless networks as an exemplary embodiment. However, the dSDN is not limited to wireless networks embodiments and it could be applied to many other network types including enterprise, wireline service providers, data centers, and Internet of Things (IoT).

[0045] Figure 1 demonstrates the network architecture of a 4G system 100. (In alternative embodiments, the system and method disclosed may also be implemented in a 2G/3G network as well). In a cellular network, the base station or BS (denoted as BTS in 2G, NodeB or NB in 3G, and eNodeB or eNB in 4G systems) is responsible for termination of a wireless link (air-interface) to the mobile device (also known as the User Equipment (UE)). The BS then transports the user data to the core network and communicates control-signaling messages with the core network. In a typical cellular network, there are thousands of BS's covering nations. Each BS covers an area known as a cell. As the demand for mobile Internet expands, the mobile network operator (MNO) needs to deploy smaller cells so it can reuse the frequency more often.

[0046] A BS is made up of the following main functional blocks: 1) radio frequency (RF) front end; 2) clock; 3) baseband; 4) power manager; and 5) central processing unit (CPU). Some BS's also have a dedicated packet processor (PP) to accelerate packet processing in hardware. BS vendors may also use custom or merchant System-on-a-Chip (SoC) or Multi-Chip-Packaging (MCP) solutions to combine various functions into a single chip (e.g., PP, CPU, and baseband).

[0047] Even though there has been attempts to open up various interfaces in the BS (such as the Open BS Architecture Initiative that defines interfaces between these functional modules in the BS), there has been very little to no efforts to unify the

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0253

programmability of the BS itself. As a result, the MNOs suffer from the following difficulties. First, the core network Internet edge (CNIE), Packet Data Gateway (PDG) and the surrounding functions are becoming extremely complicated, non-scalable, and expensive. Second, each MNO needs certain customization and feature sets. Currently, they depend on their vendors for these customizations, which could cost the MNO tremendously both financially and in regards to time-to-market (TTM); hence, hindering innovation. In addition, usually these features are put into standards or in the vendor's feature set, thus eliminating the MNO differentiation against the other MNOs. Third, if the MNO is multi-vendor in their CNIE, coordinating all the vendors to implement the same features is usually a difficult and a time-consuming effort. As a result, more add-on network appliances are introduced into the CNIE which adds to the network management complexity. Fourth, as a BS becomes Internet Protocol (IP) based, there is more visibility into the user traffic types and new innovative opportunities are missed for creative traffic shaping features, backhaul bandwidth optimization, prioritization at radio edge, power management algorithms, etc. Fifth, many features implemented in CNIE present suboptimal performance. For example, the filtering enforcement at CNIE is fundamentally inefficient since the packets have to travel all the way through the expensive air-interface, the backhaul, various other core network elements and transport networks to get to the CNIE. Sixth, the Radio Access Network (RAN) deployments are a very expensive endeavor for MNOs. The current rigid BS designs limit innovation and force the MNOs to undergo major upgrades every few years.

[0048] The problems stated above apply to many different types of networks even though here they are presented in cellular networks as a focus. In the present disclosure, the dSDN exhibits a new paradigm in the software programmability of networks that would address the problems above and enable many more advanced features.

[0049] The dSDN may be located at a network element (or system) 200 which is shown in detail in Figure 2 or in alternative embodiments the functions of the dSDN may be divided among a plurality of network elements (or systems) which are similar in structure to network element 200. Each network element 200 may comprise one or

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0254

more system control logic 202 coupled with at least one or a plurality of processor(s) 204, system memory 206, a network interface 208 (including a transceiver 208a), and input/output (I/O) devices 210. The processor(s) 204 may include one or more single-core or multi-core processors. The processor(s) 204 may include any combination of general-purpose processors and dedicated processors (e.g., graphics processors, application processors, baseband processors, etc.). System control logic 202 for one embodiment may include any suitable interface controllers to provide for any suitable interface to at least one of the processor(s) 204 and/or to any suitable device or component in a packet network in communication with system control logic 202. System control logic 202 for one embodiment may include one or more memory controller(s) to provide an interface to system memory 206. System memory 206 may be used to load and store data and/or instructions, for example, for network element 200. System memory 206 for one embodiment may include any suitable volatile memory, such as suitable dynamic random access memory (DRAM), for example. System memory 206 may also include non-volatile memory including one or more tangible, non-transitory computer-readable media used to store data and/or instructions, for example, such as the embodiments described herein with regard to the dSDN. The non-volatile memory may include flash memory, for example, and/or may include any suitable non-volatile storage device(s), such as one or more hard disk drive(s) (HDD(s)), one or more compact disk (CD) drive(s), and/or one or more digital versatile disk (DVD) drive(s). The system memory 206 may include a storage resource physically part of a device on which the network element 200 is installed or it may be accessible by, but not necessarily a part of, the device. For example, the system memory 206 may be accessed over a network via the network interface 208 and/or over Input/Output (I/O) devices 210. Network interface 208 may include a transceiver 208a to provide a radio interface for network element 200 to communicate over one or more network(s) and/or with any other suitable device. Network interface 208 may include any suitable hardware and/or firmware. The network interface 208 may include a plurality of antennas to provide a multiple input, multiple output radio interface. Network interface 208 for one embodiment may include, for example, a wired network adapter, a wireless network adapter, a telephone modem, and/or a wireless modem. For one

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0255

embodiment, at least one of the processor(s) 204 may be packaged together with logic for one or more controller(s) of system control logic 202. For one embodiment, at least one of the processor(s) 204 may be integrated on the same die with logic for one or more controller(s) of system control logic 202. In various embodiments, the I/O devices 210 may include user interfaces designed to enable user interaction with the network element or system 200, peripheral component interfaces designed to enable peripheral component interaction with the network element or system 200, and/or sensors designed to determine environmental conditions and/or location information related to the network element or system 200. In various embodiments, the peripheral component interfaces may include, but are not limited to, a non-volatile memory port, a universal serial bus (USB) port, an audio jack, and a power supply interface.

dSDN Architecture

[0050] In the dSDN architecture, the network features are virtualized and may be distributed across separate network elements that cooperate together to create an advanced, programmable, and scalable network. In an alternative embodiment, the network features may be located at a single network element. A high-level overview of a dSDN system 300 is depicted in Figure 3. The dSDN system 300 consists of a flexible network device (fxDevice) 302, a flexible cloud platform (fxCloud) 304, an application management portal (fxManager) 306, and an infrastructure application market place (fxStore) 308. The fxDevice 302 is a network device that is powered by a sandboxing operating system named flexible operating system (fxOS) 302a. The sandboxing operating system ensures each application runs as a dedicated process in sure isolation from the other applications. The fxOS 302a may be built based on an existing sandboxed operating system (OS) (e.g., Android®) by extending several aspects of such an OS with routing/networking layers and supports for operating as the wireless infrastructure. The fxDevice 302 may host several independent and securely isolated applications (on top of fxOS 302a) named here fxDeviceApp 302b.

[0051] In the dSDN system, fxDeviceApp 302b may have a sister app in the backend cloud infrastructure (i.e., flexible cloud platform 304) referenced here as fxCloudApp 304a. The fxCloudApp 304a in the cloud is paired with its fxDeviceApp 302b in the fxDevice 302. The fxCloudApp 304a and the fxDeviceApp 302b

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0256

collectively form a distributed application (dApp or fxApp). In this description, when fxDeviceApp is referenced it shall mean any application that may have software components in the fxDevice, fxCloud, or both. In general, an fxApp package may include the following software components:

fxDeviceApp binary;

fxCloudApp binary;

Manifest files; and

Signatures.

[0052] It is important to note that the fxDeviceApp 302b and fxCloudApp 304a could use any protocol to communicate with each other and there is no need for standardizing this communication allowing for ultimate freedom for the developers. This allows for the system to operate in a loosely coupled autonomous fashion allowing for asynchronous communication in a distributed fashion.

[0053] The fxApps lifecycle is managed via fxManager 306. The fxManager 306 presents a user-friendly portal to the network administrator (admin) to discover, test prior to deployment, provision, and deprovision dApps. Using fxManager 306, the admin may discover new fxApps from the fxStore 308, which presents all the tested and certified vApps as well as showing the supported fxOS version, support hardware platforms, and other information such as reviews and number of commercial deployments by all NOIT customers.

fxOS

[0054] fxDevice 302 is defined as any networking equipment embedded with a special flexible operating system (fxOS) 302a. The purpose of fxOS 302a is to enable a carrier-class programmable platform. The fxOS 302a is configured to do, but is not limited to, the following. First, since fxOS 302a is designed for network equipment, it allows for efficient execution and running all the time. A plurality of power save modes may be implemented to ensure energy saving schemes without hindering the functionality of the fxDevice 302. Second, it enables simultaneous execution of several flexible applications (fxDeviceApp 302b) on the fxDevice 302. Third, it allows for application of a legacy compiled binary or byte-code of another language (for example, Java or Android®) in addition to fxDeviceApps 302b. Fourth, it allows

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0257

FastPath processing for data plane packets. The packet forwarding in this case separated from the control plane operations. The FastPath engine allows integration with specialized hardware accelerators or packet processors (PP). The packet forwarding may be implemented with software/kernel enhancements all in software. Another FastPath implementation may use both software/kernel enhancement and hardware packet processors (PP). Fifth, the fxOS 302a is capable of Deep Packet Inspection (DPI) and identifying all the flows and their related protocols, services and present this information in a format useful to application developers. The fxOS 302a can act on the data flows as the directed by the application developer. Sixth, applications are securely isolated from each other and to communicate with them is completely configurable by policies defined by the admin. Seventh, resource utilization of each application is policed such that an application may not be able to exceed its usage. For example, an application is pre-allocated the memory size, cache/storage space, and CPU cycles. Eighth, the applications may be signed with unique certificate security keys. The security keys and certificates may be allocated and/or signed by a certificate authority. The fxDevice 302 may be protected by validating signed applications to run or to install in the device. Ninth, it allows hot upgrade of the software and applications on the platform with little to no interruption to the operational aspect of the platform and its applications. Tenth, the fxOS 302a may boot by using the secure code from the network servers (cloud-booting). This would ensure tamper-proof software foundation for the fxDevice 302. Eleventh, there may be a secure and programmable firewall in the fxOS 302a. This embedded firewall protects against attacks to the fxDevice 302 and network attached to the network. The fxOS 302a firewall may be configured by the developer and it does allow replacing of attack detection algorithms (e.g., man-in-the-middle detection algorithm). The fxOS 302a may allow secure caching of user data, web pages, and media files (video and audio) in the fxDevice 302. The secure connections to the cloud (fxCloud 304) can enable sharing and control of the caching between the fxCloud 304, fxDevice 302, or the rest of the network. Twelfth, the fxOS 302a can support multi-threading and multi-core CPUs. This feature may dynamically distribute the threads to various CPU core without developers assistance.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0258

[0055] The fxOS 302a allows secure connection to the fxCloud 304. This secure connection allows communication between fxDeviceApps 302b in the fxDevice 302 and fxCloudApp 304a in the fxCloud 304. The fxOS 302a controls the access of applications to platform APIs. An application is categorized into an Access Level (AL) based on which the fxOS 302a decides that the application does not exceed its AL and access APIs that it is not authorized. The fxCloud 304 and the fxOS 302a create a virtual fabric (fxVF) for messaging between applications. The actual message can travel directly between two fxDevices or it may traverse the fxCloud 304. The messaging complexity may be abstract for the developer.

[0056] Figure 4 demonstrates an examplary view of the software layers in the fxOS 302a. Each layer of this software stack is modified and redefined to optimize for embedded networking equipment of wireless and wireline nature. Starting from the bottom of the stack, the kernel also connects to the hardware resources via the device drivers. Examples of such resources are wireless baseband SoC 402, hardware packet processor 404, Wide Area Network (WAN) uplink interface 406, cache and storage 408, and multi-core CPU 410. The OS kernel 412 is enhanced with the FastPath function 414 to accelerate packet processing (by help of hardware features such as Intel's Data Plane Development Kit (DPDK) or pure software enhancements) and other related data path functions. The enhanced kernel with FastPath 412, 414 would enable visibility into the data flows, sniffing the Non-Access Stratum (NAS) signaling (which is the signaling between the UE and the Core Network and the RAN is supposed to be transparent to), policy enforcements, regular packet forwarding functions, collection of flow statistics into Flow Information Base (FIB), and encryption/decryption functions.

[0057] Above the kernel, there are several tools 416, native daemons 418, native libraries 420 and Hardware Abstraction Layers (HAL) 422. These tools abstract out the hardware dependencies for the upper layers and programmers. The Runtime 424 is an embedded virtual machine capable of securely isolating and executing the applications. System Services 426 are a set of services always running and available to the developers (e.g., timing and messaging services).

[0058] fx-Framework Library 428 consists of several Frameworks. The new

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0259

frameworks are the library extensions introduced explicitly for the dSDN framework. Each of these frameworks adds a set of methods of (functions) and data structures for the following examples:

Wireless Manager 428a: manipulation, monitoring, and configuration of the wireless interfaces;

System Manager 428b: configuration, provisioning, and changing major settings of the fxDevice 302;

Flow Controller 428c: allows for manipulation of data path, tunneling, switching, and optionally coordinate this activity with fxCloud 304;

Energy Management 428d: enables fine-grained control over various elements in the fxDevice 302 that could be moved into various power stares, for example: ON, OFF, SLEEP, SUSPEND;

Big Data Manager 428e: manipulation, processing, and organization of data collected from various elements. This framework could enable close coordination between the fxDevice 302 and the fxCloud 304;

Content Manager 428f: for data sharing (database sharing) between the applications within the fxDevice 302 or the fxCloud 304 and between applications in fxDevice 302 and fxCloud 304. The Content Manager 428f may use URI to address the contents to be shared. The Content Manager 428f allows access to the application data according to the policy settings and permissions;

Policy Manager 428g: responsible for execution and implementation of polices set up by the admin. The application developers could use this framework to perform queries about the permissions, limitations, and rules;

Security Manager 428h: allows for access to the security protocol libraries and rewriting some of security algorithms such as the man-in-the-middle detection algorithm;

General Framework: general compute and logic building that may be

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0260

inherited from existing general OS frameworks.

[0059] Using the frameworks, there are potentially at least two types of application types that are possible. First, the System Native Applications 430 that are provided as the initial application load into the platform. These applications could be used by other applications by links or API. Second, the fxDeviceApps 302b that are written by third party developers or the NOIT.

[0060] Figure 5 further details the FastPath 414 architecture. In this approach, the FastPath 414 is divided into the Virtual Switch (data-plane) 502 and the Switch Controller (control-plane) 504. Such separation makes possible flexible deployment wherein the function of the Switch Controller 504 could move to the fxCloud 304 by an admin decision or by the dSDN system intelligence for example in the case of failure, redundancy, stateful reboot, or Hot Upgrade. The Virtual Switch 502 may include several functions and may interact with the Switch Controller 504 in the following exemplary ways:

Firewall 502a: implements the policies given to it by the firewall logic 504a in the fxOS Daemons.

Flow/NAS Discover 502b: continuously monitors data packets and discovers new flows and NAS signaling. It would then report it to the DPI Logic 504b where the flow intelligence resides and matching to meta-data happens. DPI Logic 504b in turn updates the Flow Information Base Information (FIB) 502c of the Virtual Switch 502.

FastPath & DPI Processing 502d: forwards, routes packets, and performs DPI functions as instructed by the FIB 502c. It also collects statistics of packets and reports them to the Monitoring Logic 504c.

Policy Enforcement 502e: this function enforces various policies on the traffic such as QoS marking according to the policies received from the Policy Logic 504d.

Shaper 502f: this function could shape (queue) the traffic according to the available bandwidth on the uplink or WAN interface.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0261

Virtual Fabric (fxVF) 504e: enables transparent switching of application and system messages of a dApp between the fxCloud 304 and the fxDevice 302 and between different dApps. The fxVF 504e is an abstraction layer that hides most of the complexities of messaging from the developers. The fxVF 504e uses the policies and firewall rules to enforce security and policies.

[0061] The fxOS 302a is mostly used for networking applications. There are some potential use cases where there is a need for a general purpose OS to run the legacy software applications with no need of recompilation. An example of a double OS situation may be where a digital signage (using a general purpose OS such as Android®) in a shopping mall also acts as a smart small-cell (using the fxOS 302a). In such a solution, any existing Android® application (for example) could be loaded in the fxDevice 302 and in parallel networking applications (fxDeviceApps 302b) could run on the same fxDevice 302.

[0062] Figure 6 presents a potential software layer of the fxOS 302a co-existing alongside a General Purpose OS 602. Another implementation may be a standalone fxOS without the company of another general OS. It is important to note that the shown co-existence allows reuse of the kernel so there would be no need to re-port the drivers of hardware resources twice.

fxCloud

[0063] The fxCloud 304 is an integral part of the dSDN framework. The fxCloud 304 connects to various parts of the fxDevice 302 to form a single virtual view of the system. Examples of the main fxCloud 304 responsibilities include the following. First, it interacts and manages all applications, firmware, and fxOS 302a in the fxDevice 302. Second, it secures isolation environments for distributed apps. Third, it directs the Virtual Machine lifecycle management including load-linked VM creations and destruction (Cloud Breathing). Fourth, the fxCloud 304 executes instructions defined by the fxManager 306. Fifth, it implements uniform policy (including security) distribution and execution across dSDN 300 (including the fxDevice 302, fxDeviceApps 302b, fxCloudApps 304a, and dApps). Sixth, Plug-and-Play commissioning of the fxDevice 302 introduction

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0262

and network expansion. Seventh, Distributed Resource Service (dRS) by which distributed applications can seamlessly communicate and message to each other using published APIs or proprietary methods within the pre-defined policies. Eighth, security violation discovery and alert system. Ninth, translation and/or exposure of Resources APIs (RAPI) to the systems outside the dSDN 300 within pre-defined policies. Tenth, fxCloud 304 in coordination with the fxDevice 302 creates a Distributed Notification Service (dNS) across the dSDN 300 by which the applications could be notified of an event to wakeup and/or respond to the event. Eleventh, fxCloud 304 in coordination with the fxDevice 302 creates a Distributed Content Management Service (dCMP) which provides a virtualized and distributed database system for applications to share application-specific or system-specific information with the pre-defined policies. Twelfth, the Switch Controller 504 function in the fxCloud 304 could take control of the controller function of the fxOS 302a in the fxDevice 302. This would allow controller switching from local to remote and back. Examples of such a configuration is a temporary reason where the fxOS Switch Controller 504 may not be available during reboot or software upgrade.

[0064] In the discussion below, these services and features are expanded upon.

fxDeviceApp-0 & fxCloudApp-0

[0065] The fxDevice 302 and fxCloud 304 have embedded System Applications categorized as fxDeviceApp-0 and fxCloudApp-0, respectively. The fxDeviceApp-0 is responsible for all default and core functional capabilities of the fxDevice 302 as listed below:

Firmware of the fxDevice;

Default communication protocols (e.g. LTE S1-AP, baseband software, security protocols, GTP, DNS, tunneling protocols);

Message communication protocol between the fxDevice and fxCloud;

Initial fxDevice settings;

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0263

Boot Loader;

fxDevice X.509 digital certificate;

OS and Kernel upgrades;

Upgrade and commissioning agents for the fxDeviceApp-0 upgrades and related procedures;

Establishing and initiating the message communication protocol between the fxDevice and fxCloud;

General fxDeviceApp install and upgrades;

Access to fxDevice hardware and software information.

The fxCloudApp-0 is responsible for all default and core functional capabilities of the fxCloud 304 such as:

Upgrade and commissioning agents for the fxCloudApp-0 upgrades and related procedures;

Hypervisor upgrades;

Virtual Machine lifecycle management (e.g. creation, suspension, destruction);

fxCloud X.509 digital certificate management;

OS and kernel upgrades;

General fxCloudApp install and upgrades;

Default communication protocols (e.g. security protocols, DNS, tunneling protocols).

Each software component can be managed and updated individually. To simplify the management of these software components, each software components can be sub-numbered such as fxDeviceApp-0.1, fxDeviceApp-0.2, fxCloudApp-0.1, fxCloudApp-0.2, etc.

fxManager

[0066] The network administrator may use the fxManager 306 to control much of the dSDN system 300. The fxManager portal could use HTML5, for example, as

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0264

a frontend technology. The major functions of the fxManager 306 may include the following.

fxDevice Management:

Commissioning (plug-and-play), de-commissioning, firmware/OS upgrading/downgrading;

Suspend/resume functions;

Unification of zones and partitions for example in case of network merger;

Detailed management of fxDevices (remote secure login) where the administrator may zoom into a particular or a group of fxDevices to monitor and set management alarms;

Security policy setting.

Application lifecycle management: fxManager 306 may work, coordinate, and/or perform these tasks in conjunctions with fxCloud 304 for:

Provisioning, de-provisioning, upgrade/downgrade applications. Some applications may only have a fxDevice component (Device-Centric Apps), may only have fxCloud component (Cloud-Centric Apps), or may have both fxDevice and fxCloud components (Hybrid Apps);

Creating a Test Network Environment (TNE) where the administrator can test certain configuration and applications without impacting the production networks;

Uploading applications developed by the NOIT or their partners into the Application Repository (AR). The AR is placeholder where the application packages are downloaded and stored for testing and verification prior to installation into the live dSDN system. An application could have several components (e.g. fxDeviceApp binaries, fxCloudApp signatures, manifest files, signatures) bundled together to form the Distributed Application Package (dAP);

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0265

Browsing through the certified applications in the fxStore and downloading them to the AR;

Application license accounting services by which a secure and accurate accounting of the installed applications are performed regularly. This inventory management function allows the administrator and the 3rd party ISV to account for live and active instants of the software in the network;

Application review tool where the network admin could use to review feedbacks and comments of other NOIT (maybe anonymous reviews) and could input his/her reviews for others (maybe anonymous reviews).

fxCloud Management:

Zoning and partitioning the network for administrative purposes. Zones and partitions could optionally put into different Virtual Machine (VM);

VM load management and Cloud Breathing policies;

The configuration conflict resolutions. Some configurations applied to the resources may conflict with each other (e.g., one firewall rule may state drop Netflix packets explicitly and another rule may state allowing Netflix explicitly). The fxCloud resource manager could monitor these configurations and discover the conflicts and signal those to the network administrator.

Security Management:

Security policy setting;

Inter-App communication setting;

Security of communication between fxDevice and fxCloud;

API Access Level (AL) definitions and (re-)assignment to the applications (permissions);

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0266

Risk Analysis of the applications;

Certificate and signature verification of the applications;

Content sharing between applications and associated policies.

[0067] An fxManager Portal per the above description may have an "Account" button which can be used to create admin users and their privileges. The fxManager Portal could also include a geographical map illustrating locations of fxDevices and their status. The fxManager 306 functions could be integrated into other network management tools. The fxManager 306 could potentially build the inventory of the network devices (fxDevices) showing all the available devices and statistics. As described above, the network administrator could optionally procure the applications a third party hosted application store (fxStore). The fxStore 308 could potentially categorize the applications several groups (as shown in Figure 7 which is an example view of the fxStore portal). The fxStore 308 could also show the reviews of the advertised applications. In order to maintain quality, the third party fxStore 308 may perform rigorous testing of the applications and accompany those results with the advertised application.

fxSDK

[0068] The flexible Software Development Kit (fxSDK) may include a development environment and other tools to facilitate development, testing, debugging and verifications of applications for dSDN environments. The fxSDK is unique in several aspects exemplified as follows. First, since the dSDN applications may have a device component (fxDevice 302) and cloud component (fxCloud 304), the fxSDK allows the developer to develop the components together to simplify the development and testing. Second, the dRMS offers various system level APIs with remote accessibility. In other words, the APIs in the fxDevice 302 could be accessed via the applications in the fxCloud (fxCloudApps) and vice versa. The fxSDK simplifies the usage system created APIs and application created APIs. Third, it is essential for the dSDN to ensure quality and security of applications in the network. The fxSDK may be accompanied with an extensive Risk Analysis Utility (RAU) which verifies all the APIs that are being used by the application and based on its potential danger to the

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0267

network, it would show the risk analysis and may make specific suggestions to reduce the risk. Once completed, the risk analysis result is included in the dAP for posting to the fxStore and consumed by the fxManager user (the admin). Fourth, the fxSDK would also ensure that the applications do not exceed their planned memory and resource usage. Fifth, the Integrated Development Environment (IDE) may provide various tools to actively demonstrate errors, defined APIs, security dangers, and excessive resource usage. Sixth, the IDE could also provide a flexible Simulator (fxSimulator) for the developer test the application in a network setup in presence of other network elements.

Deployment Examples

[0069] The dSDN 300 offers flexible deployment options. In this section, a few exemplary embodiments are presented.

[0070] Figure 8 demonstrates a shared cloud deployment where several NOITs (802, 804, 806) share an fxCloud 808. In order to ensure complete isolations, each NOIT may be assigned a separate Virtual Machine (VM) 802a, 804a, 806a. A good example of this deployment option may be when multiple service providers outsource their fxCloud deployment to a third party vendor that offers a SaaS/PaaS solution. It is important to note that in this option, each NOIT should have its own secure instance of the whole system. This option could potentially create a full outsourcing model for network operations including possibilities of seamless merging of networks of two or more NOITs.

[0071] Another deployment option may be a Zoned Deployment as shown in Figure 9. In this option, a large NOIT may partition their network into a plurality of zones (Zone #1, Zone#2,… Zone#n) for administrative, software compatibility grouping, or other reason. For example, it is possible that in a large network deployment, there would be different versions of fxOS with varying capabilities in the network (the older hardware may not support newer fxOS version). In this case, the service provider may decide to group the fxDevices per fxOS versions for management simplifications.

[0072] Figure 10 is another embodiment deployed in a Programmable & Virtualized Cellular Network. In this example, the dSDN is applied to the cellular

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0268

networks (e.g., LTE). As shown, the fxDevice 1002 here is the LTE eNB (4G base-station) and the fxCloud 1004 hosts the LTE core network (EPC) functions such as MME, S-GW, and P-GW in separate VMs. Within each EPC function, new features could be added using the dSDN framework. For example, a new application to optimize signaling for Machine-2-Machine (M2M) devices could be loaded into the fxDevice 1002 and the MME function in the fxCloud 1004.

PRINCIPAL PROCEDURES & SERVICES

[0073] The dSDN framework could offer extensive services and capabilities that would simplify the programmability of the network. In this section below, a set of main procedures and services are presented as further exemplary embodiments.

fxDevice Commissioning Provisioning

[0074] Figure 11 illustrates a signaling flow chart for a fxDevice Plug-and-Play Commissioning Procedure. This procedure enables fxDevice commissioning and initial setup to provide a seamless and Plug-and-Play (PnP) deployment. The fxDevice relies on the fxDeviceApp-0 and the fxCloudApp-0 to connect to the backend management system and perform the commissioning procedure. In step 1101, the fxDevice 1120 tries to discover the fxManager 1130 using the fxManager Fully Qualified Domain Name (FQDN). The fxDevice 1120 is configured to always look for the same FQDN. If the NOIT has a private cloud deployment, the DNS server in the NOIT will be configured to point to its own fxManager IP address. If the cloud is shared, the NOIT will configure their DNS server to point to the shared cloud IP address. In step 1102, the fxDevice 1120 receives the IP address of the fxManager 1130. In step 1103, the mutual authentication may be performed where the fxDevice 1120 authenticates the fxManager 1130 and vice versa. In step 1104, a secure tunnel may be created between the fxDevice 1120 and the fxManager 1130. In step 1105, the fxDevice 1120 may optionally request for the latest firmware/OS version. In this phase, the fxDevice 1120 informs the fxManager 1130 of its hardware information (e.g. vendor, model number, CPU model/speed, memory/cache size) and software information (e.g., current firmware version, current OS version, fxDeviceApp-0 version package stating versions of fxDeviceApp-0 components). In step 1106, the fxManager 1120 may send the latest firmware/OS version (or point

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0269

the fxDevice to the right URL to download the latest load). In step 1107, if a new firmware/OS load is downloaded and verified, the fxDevice 1120 installs it. In step 1108, the fxDevice 1120 requests the default settings and the default application package (the initial applications that the NOIT admin would like to install in fxDevice 1120 prior to other applications). In step 1109, the fxManager 1130 prepares the fxCloud 1140 for the new fxDevice 1120 integration into the network. In step 1110, once completed, the fxCloud 1140 informs the fxManager 1130 of its readiness. In step 1111, the fxManager 1120 sends all the default settings and the default application package to the fxDevice 1120. In step 1112, the fxDevice 1120 then sets up a secure connection to the fxCloud 1140 (more precisely the fxCloudApp-0). In step 1113, a commissioning completion message is set from the fxCloud 1140 to the fxManager 1130. In step 1114, the fxManager 1130 signal to the fxDevice 1120 the completion of the commissioning and starts th regular operation of the fxDevice 1120. At this phase, fxManager 1130 can optionally modify the settings in the security connections between fxDeviceApp-0 and fx-CloudApp-0 and between fxDeviceApp-0 and fxManager 1130.

Application Provisioning Procedure

[0075] The Application Provisioning Procedure enables provisioning and de-provisioning of applications across the dSDN system. For simplicity, the term provisioning in this disclosure is used to present all similar procedures of provisioning and de-provisioning. The application provisioning procedures are usually triggered by the network administrator and are orchestrated by the fxManager, which works in collaboration with the fxDeviceApp-0 and fxCloudApp-0.

[0076] Figure 12 shows an example procedure for Application (dApp) Provisioning. In step 1201, the network admin identifies the applications that would need to be installed in the dSDN system. These applications are already downloaded (from the fxStore or other sources) into the Application Repository (AR). Prior to installation in the live network, the authorized network admin 1220 has probably performed rigorous testing of the application in the test network. In step 1202, the fxManager 1230 performs a detailed analysis on the potential risks that this application may

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0270

create on various parts of the network and other applications. The analysis tool uses the secure manifest file and other information to discover, for example, all the APIs used by the application, the required access-level authorization level for each of those APIs, requirements on the fxVF (inter-app, intra-app communication needs), and usage of platform resources including conflict discovery and resolutions. In step 1203, the admin defines the deployment scope that the application would be applied. Examples of deployment scopes could be a given zip code, a city, an administrative domain (zone), a shopping mall, an office building, or manual handpicking of sites or fxDevices 1240. The fxManager 1230 verifies compatibility of the OS version and resouces requirements of the application and the fxDevices 1240. If the fxManager 1230 discovers incompatibility, it informs the admin and request adjustment to the deployment scope or proposes upgrading incompatible fxDevice if possible. In step 1204, the admin 1220 initiates the installation process. In step 1205, the fxManager 1230 unpacks the dApp and if there is fxDeviceApp component, it would prepare it for submission to the fxDeviceApp-0 of fxDevices in the deployment scope. In step 1206, installation commands are sent to the fxDeviceApp-0s in a secure connection already set up as part of the fxDevice Commissioning. In step 1207, the fxDevice 1240 verifies the fxDeviceApp package and performs necessary integrity checks of the received software package. Once verified, the fxDeviceApp is installed and re-verified. In step 1208, the completion of successful install is sent to the fxManager 1230 by the fxDevice 1240. In step 1209, the fxManager 1230 prepares the fxCloudApp component of the dApp (if any). In step 1210, installation commands are sent to the fxCloudApp-0s in a existing secure connection. In step 1211, the fxCloud 1250 verifies the fxCloudApp package and performs necessary integrity checks of the received software package. Once verified, the fxCloudApp is installed and re-verified. In step 1212, the completion of successful install is sent to the fxManager 1230 by the fxCloud 1250. In step 1213, the fxManager 1230 informs the admin 1220 of the successful installation process.

Hot Upgrade Procedure

[0077] The Hot Upgrade refers to a procedure by which the software upgrades

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0271

on the system have no or minimal implication on the functionalities offered by the software component subject to the upgrade process. Herein are presented two types of software upgrades. For simplicity, the use of the term upgraded in this disclosure represents both upgrade and downgrade since they both use identical processes. First, upgrade of applications on the fxDevice (i.e., fxDeviceApp) and fxCloud (i.e., fxCloudApp). Second, upgrade of core functions in the fxDevice (aka fxDeviceApp-0) or fxCloud (aka fxCloudApp-0).

[0078] The upgrade procedures for fxDevice and fxCloud are similar. Here, the focus is on the fxDevice upgrade process since it is technically more challenging due to stricter resource limitations in the fxDevice.

General Application Upgrade

[0079] The applications could be upgraded automatically or manually by the network admin. In either case, the installation of the upgrade follows a similar procedure as the Application Provisioning Procedure. Figure 13 presents the state of the machine for the application upgrade process. In state 1301, the old version of the application is still running. In Load state 1302: the new version of the application is downloaded in the local storage while the old version is still running. In Install state 1303: the new version is installed (in the memory) while the old version is still running. In Run state 1304: the new version is activated taking charge of all the related data and application state and the old version is deactivated while still remaining in the local storage. In Commit state 1305: the old version is fully removed after ensuring the new version is running properly as expected.

fxDeviceApp-0 Upgrade

[0080] The fxDeviceApp-0 is a collection of software components that perform the core functionalities of the fxDevice. As a result, this hot upgrade is quite challenging. The most important thing is to ensure that major functionalities of the fxDevice (such as packet forwarding) remain

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0272

intact during the upgrade process. The fxDeviceApp-0 upgrades may be categorized as:

- No OS Kernel upgrade; or
- OS Kernel upgrade.

No Kernel Upgrade

[0081] As described earlier, the packet forwarding function could be broken into the Virtual Switch and the Switch Controller. In order to maintain, the packet forwarding function, the Switch Controller function could be performed by another device or a second processor in the same device while the fxDevice is being upgraded. Figures 14A-14C demonstrate a few examples of distributed Switch Controller function during the upgrade. In some cases, it may be possible for the fxCloud to take over the Switch Controller function (Figure 14A), or a neighboring fxDevice (Figure 14B), or second CPU in the same fxDevice (Figure 14C). The actual software upgrade procedure would be similar to Application Provisioning Procedure and the state machines are the same as the one shown in Figure 13.

Kernel Upgrade

[0082] In this case, the packet forwarding process is mostly unavailable. Therefore, one potential solution might be for the fxDevice to redirect the entire bit stream to a neighboring fxDevice. In other words, the fxDevice would have just a bare minimum routing function working and the rest of the function would be performed by a neighboring fxDevice.

[0083] All the other upgrade procedures and state machines are similar to the ones performed for the No Kernel Upgrade process.

Virtual Fabric Service

[0084] The Virtual Fabric (fxVF) provides an abstraction layer for application to communicate with each other whether they are in the fxDevice or fxCloud. Various frameworks and services may use the fxVF service. Figure 15 illustrates fxVF 1502a, 1503a, 1504a in the fxDevices 1502 and 1503 and fxCloud 1504. The fxVF 1502a,

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0273

1503a, 1504a may use Extensible Messaging and Presence Protocol (XMPP) for this messaging as the default protocol. It is important that the developers may use their own communication protocols between the fxDeviceApps 1502b, 1503b and the associated fxCloudApp 1504b. fxVF 1502a, 1503a, 1504a provide a secure routing mechanism.

Intra-App Messaging

[0085] This is an example of where one application in the fxDevice communicates with the same instance of the app in another instance of the same application in another fxDevice. The actual messages could go directly between the fxDevice or via the fxCloud. As an example, in the case of mobile networks, this messaging could be used to transfer user specific context from one eNB to another as a user hands off to a new eNB. In LTE, Private Messages on X2 interface could be used for messaging between the eNBs that act as the fxDevices.

Inter-App Messaging

[0086] In some cases, different applications may need to communicate with each other via their published APIs. In this case, the security policies set up by the network administrator determines which applications could communicate with each other for what purpose. The fxVF follows the security policies determined by the network administrator for inter-application communications.

distributed Resources Service (dRS)

[0087] The resources could be platform resources or APIs offered by the applications. The fxCloud and fxDevice resources could include (as examples):

Firewall;
Data and statistics;
Storage (usually abundant at fxCloud);
Compute (usually abundant at fxCloud);
Load evaluator;
General settings;

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0274

Routing engine (usually relevant to the fxDevice);

Virtual Machine master controller (usually relevant to the fxCloud);

Power controller (usually relevant to the fxDevice); and

Wireless engine (usually relevant to the fxDevice).

[0088] An app (fxDeviceApp or fxCloudApp) could expose APIs to be used by the other apps. The inter-app communication is enabled by the fxVF where the policy and security provisions are enforced. The APIs exposed may be RESTful (representational state transfer) and could travel across physical network elements. Since the fxManager defines fxVF policies, the administrator could ultimately specify which APIs between which apps could communicate with each other. Figure 16 demonstrates the logical interfaces between the platform resources and applications. As shown in Figure 16, the distributed Resources Service (dRS) 1602, 1603 comprises software agents residing in the fxDevice 1604 and the fxCloud 1606. These dRS agents manage the access to the platform resolutions (including potential configuration conflict resolution) as well as facilitating inter-app APIs. The dRS 1602, 1603 allows or disallows access to platform resource or inter-app communication according to the policies defined by the network administrator using the fxManager. In summary, the dRS 1602, 1603 provides services to application developers for:

Exposing APIs to other applications;

Configuring and managing platform resource;

Policy enforcement and authorization of applications access to platform resource and other app's APIs; and

Policy conflict resolution.

distributed Content Service (dCS)

[0089] The distributed Content Service (dCS) allows the developers to seamlessly store and share the contents generated by one application with other applications and its associated application in the cloud. The dCS simplifies access to the data and brings in storage virtualization to the applications. In other words, the developers no longer would need to know where the data is actually stored (in

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0275

the cloud or on the device) and would be able to access them easily. The dCS implementation may use Virtual Fabric (fxVF) and distributed Resources Service (dRS).

distributed Notification Service (dNS)

[0090] The distributed Notification Service (dNS) is another potential tool for developers that could wake or ping an application when a particular event has occurred. For example, a load monitoring application could be notified when the CPU load on a particular fxDevice (or a target area) exceeds a certain threshold. In turn, such an exemplary application could make smart decisions on reducing the load on the CPU by forcing handoffs of users to neighboring cells (i.e. fxDevices). The dCS implementation may use Virtual Fabric (fxVF), distributed Resources Service (dRS) and distributed Content Service (dCS).

Cloud Breathing Procedures

[0091] In general, resources (compute, storage) at the cloud are more abundant. However, software licensing costs and other limitation may require smart management of resources. The Cloud Breathing, here, is defined as a mechanism where the cloud resources are automatically expanded as load increases on the system or reduced as the load decreases. This creates an automatic elasticity in the cloud dimensions. Figures 17A and 17B demonstrate procedures for the cloud breathing. Figure 17A is expansion and Figure 17B is reduction. The Load Controller in the dRS 1602, 1603 could be used to monitor the load (step 1702) on the existing VMs (e.g. CPU or memory utilization loads). The VM Master of the fxCloudApp-0 1608 could effectively act on the decisions made by the dRS's Load Controller. Again, the network administrator via the fxManager defines the policies and thresholds for such decisions.

Security Aspects

[0092] The security aspects of this solution are of outmost importance in order to ensure quality in the product networks.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0276

Identity

[0093] To create a security model, the dSDN network elements should have unique identities. The following identities may be defined:

fxDevice Identity (DID): this identity is unique globally and may be allocated at manufacturing and included in the X.509 digital certificate of the fxDevice;

fxCloud Identity (CID): this identity is unique to the NOIT domain and may be allocated by the network administrator and included in the X.509 digital certificate of the fxCloud;

fxDeviceApp Identity (DAID): this identity may be globally unique and may be allocated by the fxSDK at the point of code creation by a globally accessible server;

fxCloudApp Identity (CAID): this identity may be globally unique and may be allocated by the fxSDK at the point of code creation by a globally accessible server;

dApp Identity (AID): This identity may be globally unique and may be constructed by concatenation of DAID and CAID together;

Application Developer Identity (ADID): this is a globally unique identity allocated by a globally accessible server and may be included in the developer's X.509 digital certificate and is used to sign the final developed applications.

[0094] For addressability purposes, the identities may be presented in a URI format using the FQDN of the NOIT.

Access Level

[0095] The applications are given different access levels. The AL is used by the dRS to determine which APIs or class of APIs an application can access (e.g., a fxDeviceApp running on fxDevice should not be able normally to reboot the fxDevice). The required AL is generated by the fxSDK and included in the manifest files. It may also be published in the fxStore for that give application. A preset of ALs could be defined to categorize the applications security risks.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0277

Software Security Verification

[0096] It is important for the target network element to verify the authenticity and integrity loaded software prior to install. The uniquely defined security keys of the application developers sign the applications. The following rules may be followed to ensure software security verification:

Only the verified OS/Firmware software can install on verified hardware;

Only the verified OS/Firmware software can run on verified hardware (continuous verification of essential parts of the software);

Only the verified fxDeviceApp software can install on verified hardware;

Only the verified fxDeviceApp software can run on verified hardware (continuous verification of essential parts of the software).

[0097] Each fxDeviceApp is accompanied by a manifest that is generated by the Integrated Device Electronics (IDE) at the time of compilation of the fxDeviceApp. The manifest file specifies which libraries and frameworks the fxDeviceApp uses. The fxManager uses this information to determine the security risk of the fxDeviceApp as well as the required Access Level (AL) to run this fxDeviceApp. In order to ensure that manifest file is actually genuine manifestation of the fxDeviceApp and its integrity remains intact, a hashing algorithm could be used to generate a signature. An example is shown below:

Signature = HMAC-SHA256(K, Message);

where

Message = fxDeviceApp + Manifest

K = shared secret key between fxStore and fxManager.

[0098] The signature is generated by the IDE/fxSDK at the compile/build phase and is packaged with application binary and the manifest file. Once the distributed application package (dAP) is downloaded into the fxManager (Application Repository), the components of the signature, application binary, and manifest file are unpackaged. The fxManager then uses the same hashing algorithm to calculate

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0278

the signature as above using the pre-shared key (K). If the calculated signature and the unpackaged signature match, it would prove the authenticity of the manifest file and the integrity of both manifest file and the fxDeviceApp.

[0099] Code Generation Process (Compilation):

fxDeviceApp compiled→ Manifest file generation→ Signature calculation→ Create fxDeviceApp Package

[0100] Install Process (Consumption):

Un-package fxDeviceApp→ Signature calculation→ Signature comparison

[0101] This model works if the key K can be shared with two trusted parties (fxManager and fxStore). The manifest file that is generated by the fxSDK shall be secured by integrity checks to ensure it remains intact throughout the transaction (i.e. from compilation to consumption).

Communication Security

[0102] Security mechanisms built into the fxDeviceApp-0 and fxCloudApp-0 forces the fxDevice to only use the configured fxCloud servers for Software Security Verification, fxDeviceApp downloads, and fxDeviceApp-fxCloud communication. Optionally, the fxDeviceApp-fxCloud communication can be customized per application requirements. One implementation of this interface could use SPDY Protocol or per-application VPN.

[0103] For the purpose of inter-App communication, an Inter-App Communication (IAC) policy template is applied to an fxDeviceApp that defied the communication policies between the fxDeviceApps and fxCloudApps. The network administrator may use one of the pre-configured IAC policy templates (exemplified below):

Isolated: this type of applications cannot communicate with any other application in the same fxDevice and does not have any associated fxCloudApp;

Isolated-Extended: this type of fxDeviceApps cannot communicate with any other fxDeviceApps in the same fxDevice but could communicate with its associated fxCloudApp;

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0279

Private: a private fxDeviceApp in one fxDevice could communicate with same private fxDeviceApps in other fxDevice but does not have any associated fxCloudApp;

Private-Extended: a Private fxDeviceApps in one fxDevice could communicate with same private fxDeviceApp in other fxDevice and has its associated fxCloudApp;

Community: a group of fxDeviceApps that belong to the same community can communicate with each other;

Promiscuous: a Promiscuous app could communicate with any other Promiscuous application and Community application;

Custom: the network administrator could create custom IAC Policy templates and could include (as examples):

Template name;

Inherited pre-configured template if any;

The protocols allowed to use for communication;

The maximum data rate allowed for inter-app communication between a pair of fxDeviceApps or between an fxDeviceApps and an fxCloudApp;

Policies related to the other fxDeviceApps in the same fxDevice, in other fxDevice, associated fxCloudApp and unassociated fxCloudApps;

Policies related to direct fxDeviceApp to fxDeviceApp communication or relayed via fxCloud communication.

Secure Boot

[0104] The secure boot refers to a capability where the main boot code is in the fxCloud with the following exemplary procedure:

The Boot Loader (BL) in the fxDeviceApp-0 of the fxDevice looks for a particular DNS (DNSSEC) name of the fxCloud. The network administrator should define the DNS entry in their network. If there is a private fxCloud, the resolved domain name points to the fxCloud. If using shared cloud, the resolved

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0280

domain name points to the cloud point of presence;

A secure tunnel (TLS) is created between the BL in fxDevice and the fxCloud using pre-burned X.509 certificate;

BL identify itself to the fxCloud and include System Information;

fxCloud determines the appropriate boot file and points the BL to the exact address of the boot file in the fxCloud;

BL downloads the correct boot file;

BL runs Software Security Verification;

BL reboots the fxDevice and the system will be ready for operation.

Frameworks

[0105] The following describes an example of the API Framework for the dSDN. Some APIs are purely local to the fxDevice or the fxCloud and some may be extended from the fxDevice all the way to the fxCloud using dRS.

General Framework

[0106] This framework refers to general libraries and APIs inherited from the legacy OS. An example of the legacy OS may be Embedded Android®. This framework allows for general processing and may be used for algorithmic applications.

Wireless Framework

[0107] The Wireless Framework adds a set up functions related to managing and controlling the wireless module. Table 1 highlights some examples of methods/functions that may be available in the Wireless Framework.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0281

Table 1: Wireless Framework	
Method	Description
discoverNeighbors	Discovers wireless neighbors and returns their identities
retrieveNeighborInfo	Retrieves detail information of neighboring BS/AP information. This method may transparently call cloud servers to gather more information
measureInterference	Measures wireless interference observed on a frequency or a set of frequency channels
setWirelessConfig	Sets various parameters in the wireless module such as power, frequency channels, Multiple Input/Multiple Output (MIMO) antenna configuration, etc.
definePHYReceiverAlgorithm	Defines the physical layer receiver algorithm
connectedDevice	Returns details about the connected mobile devices (length of connection, data tx/rx) to a given wireless interface (channel, band, sector)
forceOffload	Offloads connected mobile devices to other access technologies (e.g. from cellular to WiFi)
homeAreaNetworking	Control of home automation wireless technologies such as ZigBee or Z-Wave
setNeighborCells	Sets neighboring cells information
forceHandoff	Handoffs connected mobile devices to the same access technology but different channel or frequency
topMovers	Identifies the mobile devices with highest degree of mobility

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0282

positionDevice	Accurate positioning of mobile devices which may include geo-fencing data
borrowSpectrum	Allows one BS to borrow spectrum from another
configureMACScheduler	Configures the MAC scheduler (for example prioritize a particular user)
loadBalance	Enables load balancing across radio channels
defineMACScheduler	Defines the MAC scheduler algorithm

Security Framework

[0108] The Security Framework allows the developers to use a specialized security functions/methods. Some of these new methods may use the legacy OS frameworks. Table 2 highlights some examples of methods/functions that may be available in the Security Framework.

Table 2: Security Framework

Method	Description
setDOSDetect	Detects DOS attacks on user plane as requested
configureFirewall	Provides firewall functions such as traffic filtering, permissions (this may use dRS)
defineDOSDetectAlgorithm	Allows replacing and redefining the DOS algorithm
encryptTraffic	Encrypts the content of a message or a class of traffic as requested
authenticateUser	Authenticate users as requested
integrityCheckContent	Verifies authenticity of a message using its checksum information
configuresIPS	Configures IPS functions on the user plane

FastPath Framework

[0109] The FastPath Framework allows the developers to identify and manipulate the data path with a DPI capability. Table 3 highlights some

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0283

examples of methods/functions that may be available in the FastPath framework.

Table 3: FastPath Framework	
Method	Description
matchTraffic	Identifies and matches the traffic type requested by the developer
detectCongestion	Detects if there is congestion on the data path
identifyTopURLs	Identifies the most visited URLs by the users
detectLocalConversation	Detect conversations and packet routing between the end user on the same fxDevice
detectDevice	Detects traffic of a given mobile user device
redirectTraffic	Redirect data traffic to a given network element. The traffic may be identified with a user or a class of traffic
connectedDevices	Returns all the connected mobile devices to a given fxDevice
assignAddress	Allocates a particular IP address to the mobile device
setPriority	Set traffic priority to a class of traffic
markTraffic	Marks the user traffic with the desired QoS tags
setupTunnel	Sets up data tunnel from fxDevice to a destination
Adscript	Adds a JavaScript to an active transiting web page
detectIdenticalFlows	Identifies the identical flows coming from the WAN/uplink interface on the fxDevice
aggregateFlows	Enables aggregation of flows to save WAN/uplink utilization
routeTunnel	Routes a packet or a class of traffic onto an established tunnel
identifyTopDevices	Identifies users that generate most traffic load

Messaging Framework

[0110] The Messaging Framework allows the developers to send messaging between the applications residing in the fxDevice and fxCloud. Table 4 highlights some examples of methods/functions that may be available in the Messaging Framework.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0284

Table 4: Messaging Framework	
Method	Description
sendIntraApp	Sends an Intra-App message from one fxDevice to another
sendInterApp	Sends an InterApp message from one application to another
sendfxCloud	Sends a message to the fxCloud
sendSMS	Sends an SMS to the user of choice

Caching Framework

[0111] The Caching Framework allows the developers to cache particular contents or files that are accessed frequently in the caching engine. The caching can be done locally in the fxDevice, clustered cache (shared amongst a few fxDevice), or cloud caching. Table 5 highlights some examples of methods/functions that may be available in the Caching Framework.

Table 5: Caching Framework	
Method	Description
storeForward	Enables Store and forward model for a target class of traffic where the target traffic is matched and stored upon a condition and forwarded the condition is relieved
cacheWebSite	Caches the most visited web sites
cacheVideo	Caches the most viewed videos
cacheCloud	Caches the requested content or traffic type in the cloud storage in fxCloud

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0285

Management Framework

[0112] The Management Framework enables the developers to manage the BS platforms and perform administrative procedures. Table 6 highlights some examples of methods/functions that may be available in the Management Framework.

Table 6: Management Framework

Method	Description
upgradeFirmware	Started and executes the firmware upgrade procedure on the fxDevice
upgradeOS	Started and executes the operating system upgrade procedure on the fxDevice
remoteBootup	Performs remote boot up process where the fxDevice works closely with the fxCloud to perform the bootup
remotePower	Allows fxCloudApps to remotely turn on and off a fxDevice or elements in the fxDevice (e.g. the wireless)
gatherStatistics	Instructs fxDevice to gather specific statistics. Various commands may be consolidated for gathering statistics. Number of active users Stats of time spent by users in the cell Stats of quality perceived by the users Neighboring cell info Overall traffic passed through the system Total number of connected devices per frequency, per carrier, per site, per location Handover failure statistics Cloud aggregated data processing and full report Total traffic passed Applications used (meta-data correlated) Bandwidth consumed User's mobility pattern

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0286

setMonitoring	Allows the developer to get reports and statistics of the fxDevice traffics, errors, and neighboring environment. The reporting criteria could be set up as: 1) Period, 2) Event based (reports only if a threshold is passed; a threshold could be based on absolute numbers or deltas), 3) One-time (i.e. the information is pulled once). The exact parameters to get report on may use the same data structures used by the method “getStatistics”.

getSystemInfo	Retrieves static and dynamic system info such as: • Hardware info: vendor, model number, capability profile. GPS, Cellular, WAN interfaces, RFID/NFC • Heat temperature • Vendor Specific Information • Software info: OS version, firmware information • CPU load, per-application load (CPU, memory, accelerators) • Loaded apps and status • Memory utilization • Storage utilization • Interface status & utilization (RF, backhaul, management...)
getfxDeviceLocation	Retrieves location of the fxDevice • Location would be presented in GPS coordinates and civil address • Location enhancement: Cloud, map data, and neighbor discovery could help more accurate positioning
getPowerConsumption Rate	Retrieves current power consumption rate of the fxDevice
configureWAN	Configures the WAN settings
retrieveNeighborInfo	Retrieves neighbor information from the fxCloud
getRFIDInfo	Returns the RFID/NFC information of the fxDevice
setRFIDInfo	Sets the RFID/NFC information of the fxDevice

Extensible API Framework

[0113] This framework allows the developer to create custom APIs and make it accessible for other applications (in the fxDevice or in the fxCloud). In turn, the fxCloud could present these APIs using e.g. REST technologies (via the fxCloud

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0287

Northbound Interface) to the developer outside the dSND system. There might be limits put on the APIs exposed through the cloud to avoid potential misuse or security threats. The dRS enforces the security policies defined by the admin (via fxManager). The applications take a role of Client Application (CA) or Server Application Role (SA). The CA and SA could be distributed in the fxDevice or fxCloud. The CA makes requests and SA serves the requests.

[0114] The Extensible APIs could be categorized (as examples) into:

Read-Only Data: this type of APIs allows a CA to read data from SAs without the ability to change or request any particular action from the SAs

Read-Write Data: this type of APIs allows CA application to read data from SAs with the ability to change some of their data but no option for requesting any particular action from SAs;

Procedural: this type of APIs allows CAs to execute a particular procedure in SAs. The rights to change any data still depends on whether the API allows Read-Only Data or Read-Write Data types.

Cloud Management Framework

[0115] The Cloud Management Framework presents a collection of methods enables management of the cloud services and resource. Table 7 highlights some examples of methods/functions that may be available in the Cloud Management Framework.

Table 7: Cloud Management Framework

Method	Description
detectCloudCongestion	Detects congestions on a particular VM
deleteVM	Deletes an active VM permanently
suspendVM	Suspends an active VM but doesn't remove it
activateVM	Activates an already created VM
createVM	Creates a new VM but doesn't activate it

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0288

EXAMPLES OF DAPPS

[0116] The dSDN create an end-to-end programming platform and the possibilities of vApps are only limited by the developers' imagination. An example list of potential vApps are presented in Table 8 below.

Table 8 highlights examples of use cases possible by the Distributed Software Defined Network (dSDN)

Use	Description	Framework {Methods} used
1	Distributed SON: For example, the WiFi AP could search for the least crowded channel to minimize the interference	Wireless {discoverNeighbor, retrieveNeighborInfo, measure nterference, setWirelessConfig, loadBalance}, General Framework, Messaging {sendIntraApp}
2	BS Hot Firmware Upgrade without impact to the basic functions of the BS. For this function to support full hot upgrade without operational impact, the Communication SoC should be able to	Management {upgradeFirmware}
3	BS Hot infOS Upgrade without impact to the basic functions of the BS. For this function to support full hot upgrade without operational impact, the full platform should have dual processor.	Management {upgradeOS}
4	Secure network---based booting. This allows to keep the code securely in the cloud to avoid tampering by the ODM/CM or Man---in---the---Middle attacks	Management {remoteBoot}
5	Turning on or off APs if nobody is in the office. Sensors in the building send information to fxCloudApp which in turn orders power control of the APs	Management {remotePower}
6	Store and forward model for low priority M2M applications where the target traffic is matched and stored in the case of network congestion and forwarded once the network congestion is relieved.	DataPath {matchTraffic, detectCongestion}, Cache {storeForward}
7	Replacing MAC Scheduler	Wireless {defineMacScheduler}
8	Replacing Receiver Algorithms	Wireless {definePHYReceiverAlgorithm}
9	Caching most visited website or video clip in the enterprise	DataPath {identifyTopURL}, Cache {cacheWebSite}
10	The most viewed videos and webcasts are cached in the wireless edge	DataPath {identifyTopVideos}, Cache {cacheVideo}
11	Local Routing: data and voice sessions between two parties on the same cell are routed within the same cell bypassing the core network	DataPath {detectLocalConversation, RouteLocal}
12	Local Routing: data and voice sessions between two parties on the nearby cells are routed locally bypassing the core network	DataPath {detectConversationDetect, routeLocal}
13	Temporary redirection of a data flow for monitoring and security reasons. In this use case, a particular user's traffic is tracked by redirection to a monitoring station. If user is mobile, such profile of tracking and associated redirection is transferred from fxDevice to fxDevice either directly or via the fxCloud	DataPath {detectDevice, matchTraffic, redirectTraffic}
14	Power Calendar: On---demand coverage (triggered by passing traffic thresholds in certain cells or by calendaring): Operators need to design their network for peak usage rates at high costs, leaving their network underutilized most of the time	DataPath {detectCongestion}, Management {remotePower}
15	DOS attack recognition and action could be blocking the user, the app, and/or notifying the user via an SMS message	Security {setDOSDetect, configureFirewall}, Messaging {sendfxCloud}
16	Applying certain ACLs to the BS. For example, the students in the class can only access Facebook between the breaks and access to Facebook is blocked during the class	Security {configureFirewall}
17	Forced offloading of some cellular users (users that have attached to the same cell for a while) to WiFi by changing "something" in the cellular connection to force WiFi connection (Heterogeneous)	Wireless {identifyConnectedDevice, forceOffload}
18	Service continuity and anchoring at eNB for WiFi using the same LTE IP address (Heterogeneous)	DataPath {identifyConnectedDevices, assignAddress}
19	Graphic/general processing for special use cases such as City BS	General Frameworks
20	Application filtering/throttling (secondary conditions: time---based, location---based, subscriber---type, none)	DataPath {matchTraffic}, Security {configureFirewall}
21	Local Services: The enterprise user can discover Bonjour or UPnP devices and services (e.g. printer) even when connected to LTE/3G small---cell in the enterprise. In this case, the small cell or fxDevice pro---actively listens to Bonjour and UPnP discovery and advertisement messages (SSDP) and cache the available services until a device request for such information. In the case of Bonjour, fxDevice could listen to Multicast DNS (mDNS) messages sent over cellular connection (LTE/3G) and multicast them on the LAN or WLAN interfaces.	DataPath {matchTraffic, routeLocal}
22	Local Services: The premise---owner (e.g. airport, hotel) can provide ads under the browser for sponsored WiFi or cellular (3G/LTE) access over multi---mode small cell. The premise owner could use this service to advertise special offers or third party ads. This technique uses HTML <script> for example to add special content to the bottom of the page.	DataPath {matchTraffic, addScript}

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0289

23	Distributed Content Distribution Network (CDN). In this use case, the fxDevices function as CDN that bring the contact closer to the end users.	DataPath {matchTraffic}, Cache {cacheVideo}
24	In case of national security, the service provider may limit the access to the network so only the law enforcement can use the cellular network. In this case, the admin defines a Target Area (TA) using an fxCloudApp in the fxCloud where the emergency lockdown needs to be applied. The IMSI/MSISDN number of law enforcement mobile devices are sent to the fxDevice in the TA which in turn enforce the policy. The policy may allow the public to just sent SMS while the law enforcement could have full access and priority to the system.	DataPath {matchTraffic, setPriority}, Security {configureFirewall}
25	Emails and web browsing monitored by the cloud services assisted by the agent in the dSDN network. In this case, the fxDevice detects certain traffic (email or web in this case) and redirects them to a cloud service for security cleansing.	DataPath {matchTraffic, redirectTraffic}
26	Analytics pre---analysis: The data collected from M2M devices could be processed (noise could be filtered out) and packaged (compressed) for consumption by cloud services	DataPath {matchTraffic}, General Framework, Messaging {sendfxCloud}
27	Certain traffic could be marked with QoS DSCP codes for further processing in the network	DataPath {matchTraffic, markTraffic}
28	Proprietary SP services and applications that run purely between the phone and the BS . For example, the SP may implement a certain application for the law enforcement to get direct feed of surveillance cameras in a target area (TA).	DataPath {matchTraffic}, Security {configureFirewall}, General Framework
29	IP---PBX implementation in the small---cell or WiFi AP for enterprise use. This would allow a simple out of a box solution. The VoIP phones connect to the fxDevice via the LAN connections. The WAN connection could connect to an fxCloudApp for PSTN calls	DataPath {matchTraffic}, General Framework
30	Network management system can add its own agent/probe to collect and consolidate data and transmit it to its cloud element using any protocol	Management {gatherStatistics, getSystemInfo, control}, General Framework
31	Application based performance monitoring on the wireless link (number of packets sent in idle mode, retransmit rate…). Such information could be shared with the app developer to optimize its application	Management {gatherStatistics}, General Framework
32	Seamless network sharing and mergers. This is the case where two Network Operators would like to share or merge their network. Using dSDN's infManager, the network administrators could merge the network administration control. This process may include setting up the desirable default fxApp packages of the network operator	fxManager Portal Service
33	Following a user to understand where it faces coverage issues or call drops. Once discovered, the system could send a message to the user acknowledging an issue and that the SP is trying to fix this. The information is uploaded to the cloud for further analysis	Management {gatherStatistics}, General Framework, Messaging {sendSMS}
34	Reporting: This allows the developer to get reports and statistics of the fxDevice traffics, errors, and neighboring environment. The reporting criteria could be set up as: 1) Period, 2) Event based (reports only if a threshold is passed; a threshold could be based on absolute numbers or deltas), 3) One---time (i.e. the information is	Management {setMonitoring}, General Framework
35	Proprietary security protocol between the application in the mobile device and the BS in enterprise deployment. This would allow SP to create enterprise---class LTE for indoor using small cells	Security {encryptContent, authenticateUser, Integrity, configureFirewall}, DataPath {RouteLocal}, General Framework
36	The femto cell at home that can't scan neighboring cells, gets a list of neighboring cells from the fxCloud or neighbors to broadcast to improve outbound handoffs	Wireless {discoverNeighbors, retrieveNeighborInfo, setNeighborCells}, Management {retrieveNeighborInfo}
37	IMS---capable femto enabling various SP services to initiate or transferred between home network and mobile devices	Messaging {sendfxCloud}, General Framework
38	Integrated smart home with multi---mode femto/WiFi router. For example, the application in the smart home router (fxDevice) allows for coordination of home physical security service (such as ADT services). With the fxCloudApp, the user could control the security status of his/her home and remotely control the setting of the	Wireless {homeAreaNetworking}, Messaging {fxCloud}, General Framework
39	Home CCTV DVR implementation in an integrated home router. The camera feeds are recorded on the same router caching engine	Cache {cacheVideo}, Messaging {sendfxCloud}, General Framework
40	Enterprise CCTV DVR implementation in an integrated router with cloud backup. The camera feeds are recorded in the cloud.	Cache {CacheVideo, CacheCloud}, Messaging {fxCloud}, General Framework
41	User traffic separation into various MPLS/LSP (integrating VPN and BS). In this case, users are classified into groups and each group is tunneled to the core network. For example, law enforcement use the encrypted tunnel while the other users use the default tunnel	DataPath {Match, SetupTunnel, RouteTunnel}
42	Dynamic spectrum allocation where a BS borrows unused spectrum from the neighboring BS in case of temporary congestions	DataPath {detectCongestion}, Wireless {discoverNeighbors, setWirelessConfig}, Management {retrieveNeighborInfo, borrowCarrier}, Messaging {fxCloud}

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0290

43	Proxy ANDSF (Access Network Discovery and Control Function) in the BS. When the UE tries to access the ANDSF, the Proxy ANDSF fxDeviceApp in the BS intercept the ANDSF messages and replies on behalf of the ANDSF. This would speed up the WiFi network discovery for example.	DataPath {matchTraffic}, General Framework, Messaging {sendfxCloud}
44	HD Audio calling: audio codecs in BS allows the phones with the right capabilities to communicate with HD Audio quality calling. If one of the mobile devices can't handle the HD quality audio, the BS performs the transcoding of the audio.	DataPath {matchTraffic}, General Framework
45	Enabling mobility on White Space Spectrum: The latest white space database will be downloaded. The mobile devices with special software applications make requests to the BS to retrieve the local available white space frequencies. This would allow the mobile device to handoff seamless between BS using the white space spectrum. The BS would advertise such a capability to the UEs. The secure communication allows peer---to---peer communication between the UE and the BS for certain applications such as White Space Channel lookup or site acquisition tools for field engineering and RF planning.	Messaging {sendfxCloud}, General Framework
46	Site acquisition tool where the technician uses a software tool on his mobile phone to interact with BSs to determine the best place to deploy the new BS. The phone app interests with fxDeviceApps in BSs and fxCloudApp. For example, fxDeviceApp performs air interface quality and gives a priority to traffic generated by for this	Messaging {sendfxCloud}, General Framework. DataPath {matchTraffic, setPriority}
47	DNS caching at BS to accelerate DNS lookup for subsequent users. This would improve the user experience by reducing response delays	DataPath {matchTraffic}, General Framework
48	Continuous SLA monitory for certain protocols, applications or general link quality for jitter, delays, bandwidth	DataPath {matchTraffic}, Management {gatherStatistics}, General Framework
49	Enterprise users can use the small cell deployed in their enterprise free of charge. The DPI engine in fxDevice classifies the traffic and identifies the enterprise private data and they would be routed locally and charging records are generated by the BS (the SP decides how to bill the enterprise user and it may decide to make such usage free	DataPath {matchTraffic, routeLocal}
50	On---premise small---cell aggregation where on fxDevice takes the responsibility of aggregating multiple small---cells on the premise and presenting them as a single BS/Node---B/eNodeB to the core network. In this case the aggregating fxDevice (AfxE) may take responsibility of local radio resource manager (RRM) or self---optimizing network (SON) server. AfxE can also implement and enforce policies concerning local service access on the premise (enterprise) as well as enable seamless handover between cellular technologies and on---premise (enterprise) local area network (LAN) for example using WiFi technology.	General Frameworks, Wireless {discoverNeighbors, retrieveNeighborInfo, setNeighborCells, setWirelessConfig}, Messaging {sendIntraApp}
51	Self adjusting fxCloud compute resources: The fxCloud creates and tears down virtual machines (VMs) based on the traffic load measured by itself or the packet processing and data plane engine. It is also possible that the VM adjustments are made based on the rush hour or pre---configured hours of the day or manually by the administrator.	CloudManagement {removeVM, createVM, detectCloudCongestion, actiavteVM, suspendVM}
52	Core Network Function Virtualization (cNFV) using distributed Software Defined Networking (dSDN): In the case, once an fxDevice is commissioned, its fxDeviceApp0 (i.e. firmware) uses predefined DNS names to discover the rest of the core network (e.g. MME, S---GW, etc.). The core network elements could be created and	General Frameworks, Messaging {sendInterApp, sendIntraApp), CloudManagement {deleteVM, suspendVM, createVM, activateVM, detectCloudCongestion}
53	Smart display and digital signage as wireless base stations: this use case allows combining the capabilities of a base station and the smart display in shopping malls, airports, and enterprises. There is a already electric power and connectivity for the smart display that the base station function could use.	General Frameworks
54	Tracking a mobile user at base station level even if the UE is in the idle mode. Currently in the idle mode, the MME, SGSN, or VLR/MSC can only know the location of the UE in with a accuracy of LA/RA/URA which is a very large area. The administrator uses the fxCloudApp and requests tracking of the user/UE (the admin may provide target area where the user/UE may be located; e.g. zip code or town name). The fxCloudApp contacts its fxDeviceApp in the fxDevices in the target area (TA) requesting information of a particular IMSI/MSISD/TMSI. Once an fxDevice responds indicating knowledge of the user location, the fxCloudApp directs that serving fxDevice to inform the fxDeviceApp in the next fxDevice of the instructions to track the user and contact fxCloudApp when they user hands off to due to its mobility. fxCloud could create user friendly interfaces and presentations to show for example the direction the user is moving, predict next location of the user, and estimate arrival of the user to a certain location	DataPath {matchTraffic}, Messaging {sendIntraApp, sendfxCloud}
55	In case of emergency, fxDevices in a target area block all public traffic and only allow law enforcement sessions using specially encryption and authentication between the UE and the fxDevice or fxCloud (depending on the requirements)	DataPath {matchTraffic, setPriority}, Security {configureFirewall, encryptTraffic, authenticateUser}, General Framework

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0291

56	SON triggered by the data traffic consumption. In this case, the mobile cell configuration is adjusted based on the current traffic of the cell. It could for example force handoff some users (high users or users moving fast) to neighboring cells.	DataPath {identifyTopDevices}, Wireless {forceHandoff}
57	Time based QoS and policy. In this use case, the infManager pushes certain policies in target area (TA). Such polices could be turning on extra capacity (frequency channels) or QoS policies and traffic prioritization on the WAN like	DataPath {matchTraffic, setPriority}, Wireless {discoverNeighbors, setWirelessConfig}
58	Fast moving users detection: in this use case, the application in the BS measures the speed of the user (based on the time was handed off to this cell until it was handed over to the next cell). This information could be used by the fxDevice or fxCloud to force handoff the mobile user to a larger cell to reduce handoff rates	General Frameworks, Wireless {identifyTopMovers, forceHandoff}
59	Emergency broadcast of information and live camera feeds. In case of emergency, the enterprise network administrator temporarily block all traffic and only allow emergency related traffic across the target area in the enterprise network. For example, the network admin broadcasts the camera feeds and alerts over the WiFi access points using multicast IP techniques. The client devices and digital sinages can listen to that particular multicast and show the camera feeds and emergency alerts	DataPath {matchTraffic, setPriority}, Security {configureFirewall}, General Framework
60	Dynamic advertising in shopping malls. The shopper would use an app from the shopping mall. When the shopper brings up or refreshes the app, the dSDN allows the shopping mall to pinpoint the exact indoor/outdoor location of the user (e.g. the shop the shopper is in) and send special coupon and ads for that store to the shopper.	General Frameworks, Wireless {positionDevice}, Messaging {sendIntraApp, sendfxCloud}, Security {configureFirewall}, DataPath {matchTraffic, redirectTraffic}
61	Smart tour guide. The tourist would use an app from the tourism board or museum. When the user brings up or refreshes the app, the dSDN allows to pinpoint the exact location of the tourist (e.g. the painting he/she is looking at) and send tour guide information (e.g. audio, video, hypertext). dSDN allows to localize the positioning and	General Frameworks, Wireless {positionDevice}, Messaging {sendIntraApp, sendfxCloud}, Security {configureFirewall}, DataPath {matchTraffic, redirectTraffic}
62	General hotspot landing page	DataPath {matchTraffic, redirectTraffic}, Security {authenticateUser}
63	Proprietary wireless mesh implementation	Wireless {discoverNeighbor, retrieveNeighborInfo, measureInterference, setWirelessConfig}, General Framework, Messaging {sendIntraApp}
64	Power and energy consumption map of the network. A real---time fxCloudApp pulls the power consumption details from each fxDevice periodicaly or on demand. The pulled data then are combined with the location information and correlated into a map to present a viewable consumption map	Management {getfxDeviceLocation, getPowerConsumptionRate}
65	Network deployment optimization. This application allows network operator to get advice for best optimal reconfiguration and/or redeployment of the network. For example, the network admin could get suggestion that based on the power consumptions and coverage perceived and spectrum utilization, it might be better to replace a particular group of smaller cells with a single macro---cell. This app could also pull information from larger OEMs based on the subject deployment scenario (the OEMs either provide this information statically to the tool or provide network APIs for use by this app e.g. using REST protocol)	Management {getfxDeviceLocation, getPowerConsumptionRate, gatherStatistics}
66	High availability. This app allows two or more fxDevice to form a redundancy cluster where they participate in statueful redundancy and load---balancing	Messaging {sendIntraApp}, General Framework
67	Bandwidth Bursting. This feature enables the dynamic adjustment of backhaul or WAN link bandwidth as the demands change in the enterprise or public wireless deployments. In this case, the fxDeviceApp in the fxDevice monitors the WAN link utilization, once a preset threshold percentage has reached, it messages its sister fxCloudApp which in turn would send a request to the fxDeviceApp to increase or decrease the WAN bandwidth. If additional backend reconfiguration is needed, the fxCloudApp performs those additional changes prior to informing the fxDeviceApp of the bandwidth	Management {getStatistics, configureWAN}
68	Wireless link quality measurement by the small---cell or WiFi AP: The wireless SoC in the small---cell or WiFi AP continuously monitors the link quality for power adjustment and handover reasons. Such information could be exposed to the app developers to create cell---level analytics and reports without the need to have test--- drives or field testing of service quality	Management {gatherStatistics}, General Framework
69	Data Path Consolidation: An application on the fxDevice discovers identical data feeds from a cloud server to two or more users. The fxDeviceApp in the fxDevice can discover this potential duplicate flows at the session establishment (via a proxy function) and it would consolidate those sessions into one on the uplink towards the cloud server. A good example of this use case is the web---based desktop sharing where different many users in the same office site get the same feed from the cloud server.	FastPath {detectIdenticalFlows, aggregateFlows}

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0292

[0117] The following references are herein specifically incorporated by reference in their entirety into this disclosure:

[1] 3GPP TS 23.002, “Network Architecture”;

[2] 3GPP TS 23.402, “Architecture enhancements for non-3GPP accesses”;

[3] 3GPP TS 23.401, “General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access”;

[4] Open Base Station Architecture Initiative (OBSAI); and

[5] SPDY Protocol (http://datatracker.ietf.org/doc/draft-mbelshe-httpbis-spdy/).

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0293

CLAIMS

1. A plurality of network elements in a packet system, said network elements comprising:

a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp;

a flexible cloud platform having at least one fxCloudApp, wherein the at least one fxDeviceApp and at least one fxCloudApp are in secure communication to form a distributed application; and

an application management portal which manages life cycle of the distributed applications.

2. The system of claim 1, wherein each of the flexible network device, flexible cloud platform and application management portal include an access network interface unit configured to send and receive communications and a processor with a memory associated with the network interface unit.

3. The system of claim 1 further comprising:

an infrastructure application market place in communication with the application management portal, said infrastructure market place capable of providing tested and certified applications to the application management portal.

4. The system of claim 1, wherein the at least one fxDeviceApp is responsible for a plurality of functional capabilities of the flexible network device and is capable of at least one of the group consisting of: updating the firmware of the flexible network device; updating default communication protocols of the flexible network device; initializing flexible network device settings; boot loading the flexible network device; providing

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0294

security digital certificates; providing OS and kernal upgrades; and upgrading and commissing agents for the fxDeviceApp upgrades and related procedures.

5. The system of claim 1, wherein the at least one fxCloudApp is responsible for a plurality of functional capabilities of the flexible cloud platform and is capable of at least one of the group consisting of: upgrading and commissioning agents for the fxCloudApp updates and related procedures; hypervisor upgrades; virtual machine and/or isolated execution environment lifecycle management; security digital certificate management; OS and kernel upgrades; and default communication protocols.

6. The system of claim 1, wherein the application management portal is capable of at least one of the group consisting of: receiving new applications from the infrastructure application market place; test said distributed application prior to deployment; provision said distributed application; and deprovision said distributed application.

7. The system of claim 1, wherein the flexible network device is an LTE eNB and the flexible cloud platform hosts an LTE core network functions in separate virtual machines from the group consisting of: MME, S-GW, and P-GW.

8. The system of claim 1, wherein the flexible network device and flexible cloud platform further comprise:
a virtual fabric which provides a secure communication layer for said at least one fxDeviceApp and said at least one fxCloudApp.

9. The system of claim 1, further comprising:
a second flexible network device which includes at least one fxDeviceApp which is the same as the at least one fxDeviceApp in the first flexible network device and where each of the fxDeviceApps can communicate directly.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0295

10. The system of claim 1, wherein the flexible network device and flexible cloud platform further comprise:

a distributed resource service (DRS) which controls access to a plurality of resources upon receiving instructions from the application management portal.

11. The system of claim 10, wherein the resources include at least one of a group consisting of: firewall; data and statistics; storage; compute; load evaluator; general settings; routing engine; virtual machine master controller; power controller; and wireless engine.

12. The system of claim 1, wherein the flexible network device and flexible cloud platform further comprise:

a distributed resource service which is capable of providing at least one service from a group consisting of: exposing APIs to other applications; configuring and managing platform resources; policy enforcement and authorization of applications access to platform resource and other app's APIs; and policy conflict resolution.

13. The system of claim 1, wherein the flexible network device and flexible cloud platform further comprise:

a distributed content service wherein contents generated by the at least one fxDeviceApp may be shared with the at least one fxCloudApp.

14. The system of claim 1, wherein the flexible network device and flexible cloud platform further comprise:

a distributed notification service wherein notice of a particular event is sent from the at least one fxDeviceApp to the at least one fxCloudApp.

15. The system of claim 10, wherein the DRS further includes a load controller; and wherein the load controller is adapted to monitor loads on the at least one fxDeviceApp and at least one fxCloudApp and effect change in accordance with thresholds received from the application management portal.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0296

16. The system of claim 1, wherein the fxDevice, at least one fxDeviceApp, flexible cloud platform, and the at least one fxCloudApp each have unique identities.

17. The system of claim 1, wherein the at least one fxDeviceApp and and at least one fxCloudApp are given an access level to determine which APIs can be accessed.

18. The system of claim 1, wherein the at least one flexible network device and flexible cloud platform are adapted to verify the authenticity and integrity of updates.

19. A plurality of network elements in a packet system, said network elements comprising:

a flexible cloud platform having a plurality of virtual machines, wherein each of the virtual machines has at least one fxCloudApp; and

a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp, wherein the at least one fxDeviceApp and fxCloudApp are in secure communication to form a distributed application.

20. A plurality of network elements in a packet system, said network elements comprising:

a flexible cloud platform having a plurality of zones, wherein each of the plurality of zones has at least one fxCloudApp; and

a flexible network device adapted to process data flows in the packet system having at least one fxDeviceApp, wherein the at least one fxDeviceApp and fxCloudApp are in secure communication to form a distributed application.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0297

ABSTRACT

The Distributed Software Defined Network (dSDN) disclosed herein is an end-to-end architecture that enables secure and flexible programmability across a network with full lifecycle management of services and infrastructure applications (fxDeviceApp). The dSDN also harmonizes application deployment across the network independent of the hardware vendor. As a result, the dSDN simplifies the network deployment lifecycle from concept to design to implementation to decommissioning.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0298

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 17/142,983

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	80		N/A	
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	350		N/A	
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	400		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	18 minus 20 =	*	x 50 =	0.00	OR		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3 minus 3 =	*	x 240 =	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	830		TOTAL	

APPLICATION AS AMENDED - PART II

		(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($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	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	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
AMENDMENT B	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.

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
17/142,983	01/06/2021	2425	830	TAAG001CONT3	18	3

CONFIRMATION NO. 2943

UPDATED FILING RECEIPT

112449
Roark IP
1438 Dahlia Loop
San Jose, CA 95126

OC000000125035813

Date Mailed: 04/21/2021

Receipt is acknowledged of this non-provisional utility 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 FIRST INVENTOR, 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 corrected Filing Receipt, including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction 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 provided that the request is grantable.

Inventor(s)

Pouya Taaghol, San Jose, CA;
Vivek Ramanna, San Jose, CA;

Applicant(s)

Big Data Federation, Inc., Santa Clara, CA;

Power of Attorney: The patent practitioners associated with Customer Number 112449

Domestic Priority data as claimed by applicant

This application is a CON of 16/900,963 06/14/2020 PAT 10893095
which is a CON of 15/836,824 12/09/2017 PAT 10686871
which is a CON of 14/295,331 06/04/2014 PAT 9843624
which claims benefit of 61/834,807 06/13/2013

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: No

Permission to Access Search Results: No

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 02/17/2021

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 17/142,983**

Projected Publication Date: Request for Non-Publication Acknowledged

Non-Publication Request: Yes

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

Distributed Software Defined Networking

Preliminary Class

725

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

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 (1-866-999-4258).

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 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 U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop

technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0303

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.
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3	2943

112449 7590 01/05/2022

Roark IP
1438 Dahlia Loop
San Jose, CA 95126

EXAMINER	
HIGA, BRENDAN Y	

ART UNIT	PAPER NUMBER
2441	

MAIL DATE	DELIVERY MODE
01/05/2022	PAPER

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.

Office Action Summary	Application No. 17/142,983	Applicant(s) Taaghol et al.	
	Examiner BRENDAN HIGA	Art Unit 2441	AIA (FITF) Status Yes

-- 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 01/06/2021.

☐ 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) 1-18 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) 1-18 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 01/06/2021 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 01/06/2021.

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

4) ☐ Other: _____.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0305

DETAILED ACTION

The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

Statutory Double Patenting

A rejection based on double patenting of the "same invention" type finds its support in the language of 35 U.S.C. 101 which states that "whoever invents or discovers any new and useful process... may obtain a patent therefor..." (Emphasis added). Thus, the term "same invention," in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the claims that are directed to the same invention so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

Claims 1-18 are rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 1-18 of prior **U.S. Patent No. 10,893,095**. This is a statutory double patenting rejection.

In addition claim 6 is rejected under 35 U.S.C. 101 as claiming the same invention as that of claim 1 of prior **U.S. Patent No. 10,686,871**. This is a statutory double patenting rejection.

Non-Statutory Double Patenting

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 double patenting rejection is appropriate where the claims at issue 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).

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 reference 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. See MPEP § 717.02 for applications subject to examination under the first inventor to file provisions of the AIA as explained in MPEP § 2159. See MPEP §§ 706.02(l)(1) - 706.02(l)(3) for applications not subject to examination under the first inventor to file provisions of the AIA. A terminal disclaimer must be signed in compliance with 37 CFR 1.321(b).

The USPTO Internet website contains terminal disclaimer forms which may be used. Please visit www.uspto.gov/forms/. The filing date of the application in which the form is filed determines what form (e.g., PTO/SB/25, PTO/SB/26, PTO/AIA/25, or PTO/AIA/26) should be used. A web-based eTerminal Disclaimer may be filled out completely online using web-screens. An eTerminal Disclaimer that meets all requirements is auto-processed and approved immediately upon submission. For more information about eTerminal Disclaimers, refer to http://www.uspto.gov/patents/process/file/efs/guidance/eTD-info-I.jsp.

Claims 1-18 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-21 of **US Patent No. 9,843,624** (*' Patent). Although the claims at issue are not identical to that of the 624' Patent, for example the current application does not require the limitations of *"processing data packets in a mobile data wireless network"* and "*wherein the programmable network device and the programmable cloud device form unified capabilities using a plurality of application programming interfaces (APIs) wherein the APIs accessible to each of the plurality of the first network applications and the plurality of second network applications are determined based on a respective access level associated with each of the plurality of first and second network applications"* (see claim 1 of the 624' Patent), the claims are not patentably distinct as the broader claim scope of the current Application is anticipated by the narrower claim scope of the 624' Patent.

Claims 1-18 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-12 of US Patent No. **10,686,871** (871' Patent). Although the claims at issue are not identical to that of the 871' Patent, for

example the current application does not require *"wherein at least one of the plurality of first network applications in the programmable network device and at least one of the plurality of second network applications in the programmable cloud device are in secure communication with each other through a virtual fabric to form a distributed application"* (see claim 1 of the 871' Patent), the claims are not patentably distinct as the broader claim scope of the current Application is anticipated by the narrower claim scope of the 871' Patent.

Allowable Subject Matter

Claims 1-18 are allowed.

The following is an examiner's statement of reasons for allowance:

For similar reasons noted in parent application 16/900963 (US Patent No. 10,893,095), the prior art does not teach or render obvious, before the earliest effective filing date of Applicant's claimed invention, in the specific combinations and manner recited within the claims, the features of:

"…a programmable network device hosting a plurality of first network applications;

a programmable cloud device hosting a plurality of second network applications;

wherein at least one of the plurality of first network applications in the programmable network device and at least one of the plurality of second network applications in the programmable cloud device are in secure communication with each other to form a distributed application;

the programmable network device and programmable cloud device are each powered by a sandboxing operating system which facilitates deployment of the plurality of first and second network applications and facilitates upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device; wherein the programmable network device verifies authenticity of the upgrades to the plurality of first network applications and the programmable cloud device verifies authenticity of the upgrades to the plurality of second network applications and wherein the verification is based on unique security keys associated with each of the plurality of first and second network applications;

an application repository storing distributed applications which have been tested for installation in the programmable network device and programmable cloud device; and

an application management portal coupled to the programmable network device, programmable cloud device and the application repository, wherein the application management portal manages usage of the distributed applications on the programmable network device and programmable cloud device."

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure (see PTO 892).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to BRENDAN HIGA whose telephone number is (571)272-5823. The examiner can normally be reached on Monday - Friday 8:30 AM - 5:00 PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at http://www.uspto.gov/interviewpractice.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wing Chan can be reached on (571) 272-7493. 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). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/BRENDAN Y HIGA/
Primary Examiner, Art Unit 2441

Notice of References Cited	Application/Control No. 17/142,983	Applicant(s)/Patent Under Reexamination Taaghol et al.	
	Examiner BRENDAN HIGA	Art Unit 2441	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20180309747-A1	10-2018	Sweet; Carson	H04L63/20	1/1
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC 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.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0312

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	Art Unit
	BRENDAN HIGA	2441

CPC - Searched*		
Symbol	Date	Examiner
H04L 9/08-0838; 29/08135, 08981; 41/08-082, 0866-0876; 45/563; 63/0218; 67/10	01/01/2022	BH

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
see SEARCH history	01/01/2022	BH
IP.com (NPL+Patents)	01/01/2022	BH
Google Scholar (see search history and results)	01/01/2022	BH

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
H04L	9/08-0838; 29/08135, 08981; 41/08-082, 0866-0876; 45/563; 63/0218; 67/10	01/01/2022	BH

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0313

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	**Art Unit**
	BRENDAN HIGA	2441

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

CLAIMS

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

CLAIM		DATE								
Final	Original	01/01/2022								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0314

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L1	92	((("TAAGHOL") near3 ("Pouya")) OR (("RAMANNA") near3 ("Vivek"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2021/12/31 03:08 PM
L2	792	distribut$3 adj2 (Software adj2 defin$3 adj2 Network (SDN)) d$1sdn	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2021/12/31 03:09 PM
L3	3	1 AND 2	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2021/12/31 03:09 PM
L4	40	("20020016954" OR "20040107417" OR "20040237081" OR "20050246703" OR "20060271341" OR "20070078988" OR "20080209491" OR "20090007091" OR "20090113327" OR "20090276771" OR "20090300057" OR "20110075611" OR "20110265077" OR "20120066665" OR "20120096071" OR "20120266156" OR "20120300615" OR "20130007254" OR "20130160147" OR "20130223442" OR "20130263104" OR "20140052976" OR "20140241158" OR "20140254373" OR "20140259012" OR "20140317293" OR "20150160877" OR "20150173121" OR "20150242237" OR "20160224460" OR "20190165997" OR "7114021" OR "7630706" OR "8132166" OR "8495611" OR "8528108" OR "8650299" OR "8756595" OR "8843933" OR "9226333").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:00 PM
L5	66354	(H04L29/08135,08981	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0315

		H04L41/08-082 H04L45/563 H04L67/10).cpc.					05:02 PM
L6	4224	H04L41/0866-0876.CPC.	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:02 PM
L7	18633	H04L9/08-0838.cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:03 PM
L8	2	5 AND 6 AND 7	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:03 PM
L9	8	5 AND 4	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:04 PM
L10	647791	SDN VNF VNFCI Kubernetes Docker openflow openstack MEC (Fog distributed mobile edge) adj5 (cloud computing)	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:05 PM
L11	26802	5 AND 10	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:05 PM
L12	174	2 AND 11	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:06 PM
L13	13	12 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:06 PM
L14	45947	(file application image container app) adj5 (portal merchant market$1place hub registry register$3 repository database DB) SAME (push$3 pull$3 download$3)	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:07 PM
L15	504	14 AND 11 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:08 PM
L16	17	15 AND (6 7)	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:08 PM
L18	140896	(file application image container app) adj5 (portal merchant market$1place hub registry register$3 repository database DB directory) SAME (push$3 pull$3 download$3)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/01/01 05:16 PM
L19	8079	18 SAME (container	(US-PGPUB; USPAT;	OR	ON	ON	2022/01/01

		tenant mult$1tenant sandbox$4 virtual adj2 machine docker)	USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)				05:17 PM
L20	1377	19 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:18 PM
L21	7	20 AND (6 7)	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:18 PM
L22	351	10 AND 20	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:20 PM
L23	81577	SDN VNF VNFCI Kubernetes Docker openflow openstack MEC (Fog edge) adj5 (cloud computing)	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:23 PM
L24	4	23 AND 22	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:23 PM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N1	66354	(H04L29/08135,08981 H04L41/08-082 H04L45/563 H04L67/10).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:10 PM
N2	4224	H04L41/0866-0876.CPC.	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:10 PM
N3	18633	H04L9/08-0838.cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:10 PM
N4	1541	N1 AND (N2 N3) AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:11 PM
N5	350083	(key token).clm.	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:12 PM
N6	227	N4 AND N5	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:13 PM
N7	509125	(container tenant mult$1tenant sandbox$4 virtual adj2 machine docker).clm.	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:13 PM

N8	29	N6 AND N7	(US-PGPUB; USPAT)	OR	ON	ON	2022/01/01 05:14 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0318

Google Scholar "distributed SDN" management SIGN IN

Articles About 47 results (0.05 sec) My profile My library

Any time
Since 2022
Since 2021
Since 2018
Custom range...
— 2013

Sort by relevance
Sort by date

Any type
Review articles

include patents
include citations

Create alert

Exploiting locality in **distributed SDN** control [PDF] acm.org
Full Text @ USPTO-STIC
S Schmid, J Suomela - Proceedings of the second ACM SIGCOMM ..., 2013 - dl.acm.org
... However, the controller may not necessarily represent a single, centralized device, but the control plane may consist of multiple controllers in charge of **managing** different administrative ...
Save Cite Cited by 235 Related articles All 20 versions

Towards an elastic **distributed SDN** controller [PDF] acm.org
Full View
A Dixit, F Hao, S Mukherjee, TV Lakshman... - ACM SIGCOMM ..., 2013 - dl.acm.org
... assumed ownership of X in the previous phase, the protocol is not complete unless A detaches from **managing** switch X. However, it cannot just detach immediately from the switch ...
Save Cite Cited by 717 Related articles All 16 versions

Towards a **distributed SDN** control: Inter-platform signaling among flow processing platforms [PDF] semanticscholar.org
Full Text @ USPTO-STIC
F Salvestrini, G Carrozzo, N Ciulli - 2013 IEEE SDN for Future ..., 2013 - ieeexplore.ieee.org
... a **management** challenge that involves significant modification to the network configuration. ... Also, most of the SDN products adopt a centralized SDN controller to **manage** the overall ...
Save Cite Cited by 11 Related articles All 3 versions

Logically centralized? State distribution trade-offs in software defined networks [PDF] acm.org
D Levin, A Wundsam, B Heller, N Handigol... - Proceedings of the first ..., 2012 - dl.acm.org
... Consequently, we ask: "How does **distributed SDN** state impact the performance of a ... the underlying **distributed SDN** state. The more complex application state **management** approach is ...
Save Cite Cited by 431 Related articles All 13 versions

A **Distributed SDN** Control Plane for Consistent Policy Updates [PDF] arxiv.org
M Canini, P Kuznetsov, D Levin, S Schmid - arXiv preprint arXiv ..., 2013 - arxiv.org
... This paper studies a **distributed SDN** control plane that enables concurrent and robust policy ... Our interface is however intended to model realistic network **management** operations, ...
Save Cite Cited by 1 Related articles All 5 versions

[PDF] Routing and State Distribution Trade-Offs in SDN [PDF] euroview2012.org
N Sarrar, D Levin, A Feldmann - euroview2012.org
... We compare two different control application approaches which operate on **distributed SDN** ... **distributed SDN** state. Alternately, the more complex application state **management** ...
Save Cite Related articles

Thales Communications & Security
K Phemius, M Bouet, J Leguay - 2013 - Citeseer
... extensible **DIstributed SDN** COntrol plane able to cope with the distributed and heterogeneous nature of modern overlay networks and wide area networks. DISCO controllers **manage** ...
Save Cite Cited by 2 Related articles

[PDF] SmartBlock-SDN: An Optimized Blockchain-SDN Framework for Resource **Management** in IoT [PDF] researchgate.net
A Mosavi, A Rahman - 2000 - researchgate.net
... [35] proposed SADFIR, an interactive clustering routing protocol that enables **distributed SDN** controllers to collaborate with forwarding network devices for routing the sensed ...
Save Cite Related articles

[PDF] Seamless interworking of SDN and IP [PDF] acm.org
Full View
P Lin, J Hart, U Krishnaswamy, T Murakami... - Proceedings of the ..., 2013 - dl.acm.org

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0319

... to make it easier to program and **manage** the networks. Transitioning the widely deployed
... The SDN AS is controlled by ONOS (an experimental **distributed SDN** Operating System) [2]...
Save Cite Cited by 127 Related articles All 11 versions

Using SDN for cloud services provisioning: the XIFI use-case Full Text @ USPTO-STIC

E Escalona, JIA Baranda, LMC Murillo... - 2013 IEEE SDN for ..., 2013 - ieeexplore.ieee.org

... In this paper we will present the architecture that XIFI plans to setup and **manage** several DC
... The provisioning manager will instruct **distributed SDN** controllers local to each XIFI DC for ...
Save Cite Cited by 11 Related articles All 3 versions

Related searches

distributed sdn **control**

blockchain sdn **resource** management

data store distributed sdn **control plane**

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0320

Web Search History

date, time	web site	search string
1/1/2022 5:30:23 PM	Google Scholar	distributed SDN or DSDN
1/1/2022 5:30:37 PM	Google Scholar	"distributed SDN"
1/1/2022 5:30:43 PM	Google Scholar	"distributed SDN" [before:2013]
1/1/2022 5:31:05 PM	Google Scholar	[before:2013] "distributed SDN" key token
1/1/2022 5:31:13 PM	Google Scholar	[before:2013] "distributed SDN" key or token
1/1/2022 5:31:32 PM	Google Scholar	[before:2013] "distributed SDN" verification or authentication
1/1/2022 5:32:07 PM	Google Scholar	[before:2013] "distributed SDN" sandbox
1/1/2022 5:32:26 PM	Google Scholar	[before:2013] "distributed SDN" management

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0321

Main Concept Document
==========
US10893095
Title: Distributed software defined networking

Concept Modifiers (2)
==========
2. More Like Text: distributed SDN

1. More Like Text: key or token

Filters (1)
==========
Publication Date: To 2013-12-31

Options
=======
De-dup: Extended Family Numbers (PAT only)
Cut-off: None
Sort: Relevance

Content: Patents + NPL (187)
==========
US Patents
US Designs
US Applications
EPO Patents
EPO Applications
China Patents
China Applications
Japan Patents
Japan Applications
Korea Patents
Korea Applications
WIPO Applications
Argentina Patents
Argentina Applications
Brazil Patents
Brazil Applications
Canada Patents
Canada Applications
Chile Patents

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0322

Chile Applications
Colombia Applications
Costa Rica Applications
Cuba Patents
Cuba Applications
Dominican Republic Applications
Ecuador Patents
Ecuador Applications
El Salvador Applications
Guatemala Applications
Honduras Applications
Mexico Patents
Mexico Applications
Nicaragua Patents
Panama Applications
Peru Applications
Trinidad & Tobago Patents
Uruguay Applications
Austria Patents
Austria Applications
Belarus Patents
Belgium Patents
Belgium Applications
Bosnia & Herzegovina Patents
Bosnia & Herzegovina Applications
Bulgaria Patents
Bulgaria Applications
Croatia Patents
Croatia Applications
Czech Republic Patents
Czech Republic Applications
Czechoslovakia Patents
Czechoslovakia Applications
Denmark Patents
Denmark Applications
Estonia Patents
Estonia Applications
EUIPO Patents
Finland Patents
Finland Applications
France Patents
France Applications
Germany Patents
Germany Applications
Great Britain Patents
Great Britain Applications
Greece Patents
Greece Applications

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0323

Hungary Patents
Hungary Applications
Iceland Patents
Iceland Applications
Ireland Patents
Ireland Applications
Italy Patents
Italy Applications
Latvia Patents
Latvia Applications
Lithuania Patents
Lithuania Applications
Luxembourg Patents
Malta Patents
Monaco Patents
Montenegro Patents
Montenegro Applications
Netherlands Patents
Netherlands Applications
Norway Patents
Norway Applications
Poland Patents
Poland Applications
Portugal Patents
Portugal Applications
Republic of Moldova Patents
Republic of Moldova Applications
Romania Patents
Romania Applications
San Marino Patents
San Marino Applications
Serbia Patents
Serbia Applications
Slovakia Patents
Slovakia Applications
Slovenia Patents
Spain Patents
Spain Applications
Sweden Patents
Sweden Applications
Switzerland Patents
Switzerland Applications
Ukraine Patents
Yugoslavia/Serbia and Montenegro Patents
Yugoslavia/Serbia and Montenegro Applications
Armenia Patents
Australia Patents
Australia Applications

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0324

Cyprus Patents
Gulf Cooperation Council Patents
Hong Kong Patents
India Patents
India Applications
Indonesia Patents
Indonesia Applications
Israel Applications
Jordan Patents
Kyrgyzstan Patents
Macao Applications
Malaysia Patents
Mongolia Patents
New Zealand Patents
Philippines Patents
Philippines Applications
Saudi Arabia Patents
Singapore Patents
Singapore Applications
Taiwan Patents
Taiwan Applications
Tajikistan Patents
Tajikistan Applications
Thailand Applications
Uzbekistan Patents
Vietnam Patents
Algeria Patents
ARIPO Patents
ARIPO Applications
Egypt Patents
Kenya Patents
Malawi Patents
Morocco Patents
Morocco Applications
OAPI Patents
South Africa Patents
Tunisia Applications
Zambia Patents
Zimbabwe Patents
EAPO Patents
EAPO Applications
Georgia Patents
Georgia Applications
Kazakhstan Patents
Kazakhstan Applications
Russia Patents
Russia Applications
Turkey Patents

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0325

Turkey Applications
IEEE Xplore Publications: IEEE Periodicals
IEEE Xplore Publications: IEEE Conferences
IEEE Xplore Publications: IEEE Standards
IEEE Xplore Publications: IEEE Early Access
IEEE Xplore Publications: SMPTE Periodicals
IEEE Xplore Publications: SMPTE Conferences
IEEE Xplore Publications: SMPTE Standards
IEEE Xplore Publications: MIT Press eBooks
IEEE Xplore Publications: Wiley-IEEE eBooks
IEEE Xplore Publications: IBM Periodicals
IEEE Xplore Publications: URSI Periodicals
IEEE Xplore Publications: VDE Conferences
IEEE Xplore Publications: Periodicals from China
IP.com Prior Art Database: The IP.com Journal
IP.com Prior Art Database: Internet Society RFC
IP.com Prior Art Database: IBM TDB Archive
IP.com Prior Art Database: Legacy Journals
IP.com Prior Art Database: Software Patent Institute
OnePetro.org: Periodicals at OnePetro.org
OnePetro.org: Conferences at OnePetro.org
Other Literature: IBM Redbooks
Other Literature: PubMed Central
Other Literature: arXiv.org

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0326

Bibliographic Data

Application No: 17/142,983

Foreign Priority claimed: ○ Yes ◉ No

35 USC 119 (a-d) conditions met: ☐ Yes ☑ No ☐ Met After Allowance

Verified and Acknowledged: /BRENDAN Y HIGA/ (Examiner's Signature) | BH (Initials)

Title: Distributed Software Defined Networking

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
01/06/2021	709	2441	TAAG001CONT3
RULE			

APPLICANTS

Big Data Federation, Inc., Santa Clara, CA,

INVENTORS

Pouya Taaghol, San Jose, CA, UNITED STATES

Vivek Ramanna, San Jose, CA, UNITED STATES

CONTINUING DATA

This application is a CON of 16900963 06/14/2020 PAT 10893095

16900963 is a CON of 15836824 12/09/2017 PAT 10686871

15836824 is a CON of 14295331 06/04/2014 PAT 9843624

14295331 has PRO of 61834807 06/13/2013

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

02/17/2021

**** SMALL ENTITY ****

STATE OR COUNTRY

UNITED STATES

ADDRESS

Roark IP

1438 Dahlia Loop

San Jose, CA 95126

UNITED STATES

FILING FEE RECEIVED

$830

PTO/SB/08a (07-09)
Approved for use through 11/30/2020. 0MB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 ,no persons are reauired to respond to a collection of information unless it contains a valid 0MB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 4

Complete if Known	
Application Number	17/142983
Filing Date	01/06/2021
First Named Inventor	Taaghol
Art Unit	
Examiner Name	
Attorney Docket Number	TAAG001CONT3

U.S.PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 7114021	9-2006	Seshadri	
		US- 7630706	12-2009	Weigand	
		US- 8132166	3-2012	DeHaan	
		US- 8528108	9-2013	Ammer	
		US- 8650299	2-2014	Huang	
		US- 8756595	6-2014	Arcilla	
		US- 8843933	9-2014	Holler	
		US- 9226333	12-2015	Steiner	
		US- 20020016954	2-2002	Charisius	
		US- 20040107417	6-2004	Chia	
		US- 20040237081	11-2004	Homiller	
		US- 20050246703	11-2005	Ahonen	
		US- 20070078988	4-2007	Miloushev	
		US- 20080209491	8-2008	Hasek	
		US- 20090007091	1-2009	Appiah	
		US- 20090113327	4-2009	Pogrebinsky	
		US- 20090276771	11-2009	Nickolov	
		US- 20090300057	12-2009	Friedman	
		US- 20110075611	3-2011	Choi	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear

Examiner Signature	/BRENDAN Y HIGA/	Date Considered	12/30/2021

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

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 2 hours 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, 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, ca/11-800-PTO-9199 (1-800-786-9199) and select option 2.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /B.Y.H/

PTO/SB/08a (07-09)
Approved for use through 11/30/2020. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no persons are reauired to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 2 of 4

Complete if Known	
Application Number	17/142983
Filing Date	01/06/2021
First Named Inventor	Taaghol
Art Unit	
Examiner Name	
Attorney Docket Number	TAAG001CONT3

U.S.PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 20120300615	11-2012	Kempf	
		US- 20130007254	01-2013	Fries	
		US- 20130160147	06-2013	Draluk	
		US- 20130263104	10-2013	Basel	
		US- 20140052976	02-2014	Marino	
		US- 20140241158	08-2014	Anthony, Jr.	
		US- 20140254373	09-2014	Varma	
		US- 20140259012	09-2014	Nandlall	
		US- 20140317293	10-2014	Shatzkamer	
		US- 20150173121	06-2015	Miklos	
		US- 20110265077	10-2011	Collison	
		US- 20120066665	03-2012	Kunze	
		US- 20120096071	04-2012	Murphey	
		US- 20130223442	08-2013	Narayanan	
		US- 20150160877	06-2015	Larkin	
		US- 20150242237	08-2015	Spivak	
		US- 20160224460	08-2016	Bryant	
		US- 20190165997	05-2019	Shaikh	
		US- 8843933	09-2014	Holler	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear

Examiner Signature	/BRENDAN Y HIGA/	Date Considered	12/30/2021

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

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 2 hours 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, 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, ca/11-800-PTO-9199 (1-800-786-9199) and select option 2.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /B.Y.H/

PTO/SB/08a (07-09)
Approved for use through 11/30/2020. 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.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 3 of 2

Complete if Known	
Application Number	17/142983
Filing Date	01/06/2021
First Named Inventor	Taaghol
Art Unit	
Examiner Name	
Attorney Docket Number	TAAG001CONT3

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 8495611	07-2013	McCarthy	
		US- 20120266156	10-2012	Spivak	
		US- 20060271341	11-2016	Brown	
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T[6]

Examiner Signature	/BRENDAN Y HIGA/	Date Considered	12/30/2021

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

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 2 hours 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, 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 (1-800-786-9199) and select option 2.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /B.Y.H/

PTO/SB/08b (07-09)
Approved for use through 11/30/2020. 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.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 4 of 4

Complete if Known	
Application Number	17/142983
Filing Date	01/06/2021
First Named Inventor	Taaghol
Art Unit	
Examiner Name	
Attorney Docket Number	TAAG001CONT3

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	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, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		McKeown, et al. "OpenFlow: Enabling Innovation in Campus Networks", March 14, 2008.	
		Li et al. "Toward Software-Defined Cellular Networks", "2012 European Workshop on Software Defined Networking".	
		"Network Functions Virtualisation", "SDN and OpenFlow World Congress", Darmstadt-Germany, October 22-24, 2012.	
		Yap et al., "OpenRoads: Empowering Research in Mobile Networks", ACM SIGCOMM Computer Communication Review, 2010.	
		Bansal et al., "OpenRadio: A Programmable Wireless Dataplane", HotSDN'12, August 13, 2012, Helsinki, Finland.	
		Stallings et al., "Software-Defined Networks and OpenFlow", The Internet Protocol Journal, vol. 16, No. 1, March 2013.	
		Luo et al., "Software Defined Networking across Distributed Datacenters over Cloud", IEEE Int. Conference on Cloud Computing Tech, 2013	
		Chappell, David, "A Short Introduction to Cloud Platforms", Aug. 2008.	

Examiner Signature	/BRENDAN Y HIGA/	Date Considered	12/30/2021

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.
This collection of information is required by 37 CFR 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 2 hours 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, 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 (1-800-786-9199) and select option 2.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /B.Y.H/

Doc Code: DIST.E.FILE **Document Description: Electronic Terminal Disclaimer - Filed**	PTO/SB/26 U.S. Patent and Trademark Office Department of Commerce

Electronic Petition Request	**TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT**
Application Number	17142983
Filing Date	06-Jan-2021
First Named Inventor	Pouya Taaghol
Attorney Docket Number	TAAG001CONT3
Title of Invention	Distributed Software Defined Networking

☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action

☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.

Owner	Percent Interest
Big Data Federation, Inc.	100%

The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

9843624

10686871

10893095

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:
- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

(•) Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request. ( ) I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.	
Applicant claims the following fee status: (•) Small Entity ( ) Micro Entity ( ) Regular Undiscounted I hereby declare that all statements made herein of my own knowledge are true and that all statements made 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 so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.	
THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES I certify, in accordance with 37 CFR 1.4(d)(4) that I am: (•) An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application Registration Number 40394 ( ) A sole inventor ( ) A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application ( ) A joint inventor; all of whom are signing this request	
Signature	/Christopher J. Horgan/
Name	Christopher J. Horgan

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

Electronic Patent Application Fee Transmittal

Application Number:	17142983
Filing Date:	06-Jan-2021
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Filer:	Christopher James Horgan
Attorney Docket Number:	TAAG001CONT3

Filed as Small Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
STATUTORY OR TERMINAL DISCLAIMER	2814	1	170	170
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**170**

Doc Code: DISQ.E.FILE
Document Description: Electronic Terminal Disclaimer – Approved

Application No.: 17142983

Filing Date: 06-Jan-2021

Applicant/Patent under Reexamination: Taaghol

Electronic Terminal Disclaimer filed on February 13, 2022

☒ APPROVED

This patent is subject to a terminal disclaimer

☐ DISAPPROVED

Approved/Disapproved by: Electronic Terminal Disclaimer automatically approved by EFS-Web

U.S. Patent and Trademark Office

Electronic Acknowledgement Receipt	
EFS ID:	44983208
Application Number:	17142983
International Application Number:	
Confirmation Number:	2943
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Customer Number:	112449
Filer:	Christopher James Horgan
Filer Authorized By:	
Attorney Docket Number:	TAAG001CONT3
Receipt Date:	13-FEB-2022
Filing Date:	06-JAN-2021
Time Stamp:	21:37:35
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$170
RAM confirmation Number	E20222CL37281015
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Terminal Disclaimer-Filed (Electronic)	eTerminal-Disclaimer.pdf	41535 091953ae367ee5a6819fb0debd3e4e2ee95b2f2d	no	2
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	38107 723903ecfb649c580f89cab899ab5f814a639be4	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	79642		

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.

Electronic Acknowledgement Receipt

EFS ID:	44983238
Application Number:	17142983
International Application Number:	
Confirmation Number:	2943
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Customer Number:	112449
Filer:	Christopher James Horgan
Filer Authorized By:	
Attorney Docket Number:	TAAG001CONT3
Receipt Date:	13-FEB-2022
Filing Date:	06-JAN-2021
Time Stamp:	21:56:10
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	TAAG001CONT3_RESPONSE____.pdf	30123 554ff256c63997a5b42be3d6d8fa643668c3b5fc	no	2

Warnings:

Information:	
Total Files Size (in bytes):	30123

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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Taaghol
Serial No.: 17/142,983
Filed: 1/6/2021
Attorney Docket No.: TAAG001CONT3

Examiner:
Art Unit:

For: DISTRIBUTED SOFTWARE DEFINED NETWORKING

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

Dear Sir:

RESPONSE

Applicants offer the following election in response to the Office Action (hereinafter "Office Action") mailed on January 5, 2022.

Remarks begin on page 2 of this paper.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0341

REMARKS

Claims 1-18 are currently pending.

DOUBLE PATENTING REJECTIONS

Claims 1-18 are rejected on the ground of nonstatutory double patenting as being unpatentable over U.S. Patent No. 9,843,624. Claims 1-18 are rejected on the ground of statutory double patenting as being unpatentable over U.S. Patent No. 10,686,871 and 10,893,095.

Accordingly, Applicant has filed a Terminal Disclaimer with this response.

CONCLUSION

Therefore it is respectfully submitted that claims 1-18 are allowable over the prior art. No new matter has been added in the new claims and/or limitations presented in this Amendment.

Respectfully submitted,

ROARK IP

By: /Christopher J. Horgan/

Christopher J. Horgan
Registration No. 40,394
1438 Dahlia Loop
San Jose, CA 95126
Telephone: (408) 687-8306

Date: February 13, 2022

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0342

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.
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3	2943

112449 7590 04/08/2022

Roark IP
1438 Dahlia Loop
San Jose, CA 95126

EXAMINER	
HIGA, BRENDAN Y	

ART UNIT	PAPER NUMBER
2441	

MAIL DATE	DELIVERY MODE
04/08/2022	PAPER

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.

Office Action Summary	Application No. 17/142,983	Applicant(s) Taaghol et al.	
	Examiner BRENDAN HIGA	Art Unit 2441	AIA (FITF) Status Yes

-- 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 02/13/2022.

☐ 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) 1-18 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) 1-18 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

This Office action is in response to Applicant's request for reconsideration filed on February 13, 2022.

Claims 1-18 are pending.

Response to Arguments

Applicant's arguments filed February 13, 2022 have been fully considered but they are not persuasive.

Applicant is reminded that the filing of a terminal disclaimer **cannot** overcome a statutory double patenting rejection based upon 35 U.S.C. 101.

With that said, upon further consideration, the Examiner has withdrawn the statutory double patenting rejection of claim 6 in view of claim 1 of prior U.S. Patent No. 10,686,871.

Statutory Double Patenting

A rejection based on double patenting of the "same invention" type finds its support in the language of 35 U.S.C. 101 which states that "whoever invents or discovers any new and useful process... may obtain a patent therefor..." (Emphasis added). Thus, the term "same invention," in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the claims that are directed to the same invention so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

Claims 1-18 are rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 1-18 of prior **U.S. Patent No. 10,893,095**. This is a statutory double patenting rejection.

Allowable Subject Matter

Claims 1-18 are allowed over the prior art.

Conclusion

THIS ACTION IS MADE FINAL. 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 mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to BRENDAN HIGA whose telephone number is (571)272-5823. The examiner can normally be reached on Monday - Friday 8:30 AM - 5:00 PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at http://www.uspto.gov/interviewpractice.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wing Chan can be reached on (571) 272-7493. 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). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/BRENDAN Y HIGA/
Primary Examiner, Art Unit 2441

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	**Art Unit**
	BRENDAN HIGA	2441

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

CLAIMS

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

CLAIM		DATE								
Final	Original	01/01/2022	04/07/2022							
	1	✓	✓							
	2	✓	✓							
	3	✓	✓							
	4	✓	✓							
	5	✓	✓							
	6	✓	✓							
	7	✓	✓							
	8	✓	✓							
	9	✓	✓							
	10	✓	✓							
	11	✓	✓							
	12	✓	✓							
	13	✓	✓							
	14	✓	✓							
	15	✓	✓							
	16	✓	✓							
	17	✓	✓							
	18	✓	✓							

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0348

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	Art Unit
	BRENDAN HIGA	2441

CPC - Searched*		
Symbol	Date	Examiner
H04L 29/08135, 08981; 41/08-082, 0866-0876; 45/563; 63/0218; 67/10, 34	04/07/2022	BH

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
see SEARCH history	04/07/2022	BH
IP.com (NPL+Patents)	04/07/2022	BH
Google Scholar (see search history and results)	04/07/2022	BH

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
H04L	H04L 29/08135, 08981; 41/08-082, 0866-0876; 45/563; 63/0218; 67/10, 34	04/07/2022	BH

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0349

Main Concept Document
==========
US10893095
Title: Distributed software defined networking

Concept Modifiers (3)
==========
3. More Like Text: key or token

2. More Like Text: upgrade

1. More Like Text: orchestrate

Filters (1)
==========
Publication Date: To 2013-12-31

Options
==========
De-dup: Extended Family Numbers (PAT only)
Cut-off: None
Sort: Relevance

Content: Patents + NPL (187)
==========
US Patents
US Designs
US Applications
EPO Patents
EPO Applications
China Patents
China Applications
Japan Patents
Japan Applications
Korea Patents
Korea Applications
WIPO Applications
Argentina Patents
Argentina Applications
Brazil Patents
Brazil Applications
Canada Patents

Canada Applications
Chile Patents
Chile Applications
Colombia Applications
Costa Rica Applications
Cuba Patents
Cuba Applications
Dominican Republic Applications
Ecuador Patents
Ecuador Applications
El Salvador Applications
Guatemala Applications
Honduras Applications
Mexico Patents
Mexico Applications
Nicaragua Patents
Panama Applications
Peru Applications
Trinidad & Tobago Patents
Uruguay Applications
Austria Patents
Austria Applications
Belarus Patents
Belgium Patents
Belgium Applications
Bosnia & Herzegovina Patents
Bosnia & Herzegovina Applications
Bulgaria Patents
Bulgaria Applications
Croatia Patents
Croatia Applications
Czech Republic Patents
Czech Republic Applications
Czechoslovakia Patents
Czechoslovakia Applications
Denmark Patents
Denmark Applications
Estonia Patents
Estonia Applications
EUIPO Patents
Finland Patents
Finland Applications
France Patents
France Applications
Germany Patents
Germany Applications
Great Britain Patents
Great Britain Applications

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0351

Greece Patents
Greece Applications
Hungary Patents
Hungary Applications
Iceland Patents
Iceland Applications
Ireland Patents
Ireland Applications
Italy Patents
Italy Applications
Latvia Patents
Latvia Applications
Lithuania Patents
Lithuania Applications
Luxembourg Patents
Malta Patents
Monaco Patents
Montenegro Patents
Montenegro Applications
Netherlands Patents
Netherlands Applications
Norway Patents
Norway Applications
Poland Patents
Poland Applications
Portugal Patents
Portugal Applications
Republic of Moldova Patents
Republic of Moldova Applications
Romania Patents
Romania Applications
San Marino Patents
San Marino Applications
Serbia Patents
Serbia Applications
Slovakia Patents
Slovakia Applications
Slovenia Patents
Spain Patents
Spain Applications
Sweden Patents
Sweden Applications
Switzerland Patents
Switzerland Applications
Ukraine Patents
Yugoslavia/Serbia and Montenegro Patents
Yugoslavia/Serbia and Montenegro Applications
Armenia Patents

Australia Patents
Australia Applications
Cyprus Patents
Gulf Cooperation Council Patents
Hong Kong Patents
India Patents
India Applications
Indonesia Patents
Indonesia Applications
Israel Applications
Jordan Patents
Kyrgyzstan Patents
Macao Applications
Malaysia Patents
Mongolia Patents
New Zealand Patents
Philippines Patents
Philippines Applications
Saudi Arabia Patents
Singapore Patents
Singapore Applications
Taiwan Patents
Taiwan Applications
Tajikistan Patents
Tajikistan Applications
Thailand Applications
Uzbekistan Patents
Vietnam Patents
Algeria Patents
ARIPO Patents
ARIPO Applications
Egypt Patents
Kenya Patents
Malawi Patents
Morocco Patents
Morocco Applications
OAPI Patents
South Africa Patents
Tunisia Applications
Zambia Patents
Zimbabwe Patents
EAPO Patents
EAPO Applications
Georgia Patents
Georgia Applications
Kazakhstan Patents
Kazakhstan Applications
Russia Patents

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0353

Russia Applications
Turkey Patents
Turkey Applications
IEEE Xplore Publications: IEEE Periodicals
IEEE Xplore Publications: IEEE Conferences
IEEE Xplore Publications: IEEE Standards
IEEE Xplore Publications: IEEE Early Access
IEEE Xplore Publications: SMPTE Periodicals
IEEE Xplore Publications: SMPTE Conferences
IEEE Xplore Publications: SMPTE Standards
IEEE Xplore Publications: MIT Press eBooks
IEEE Xplore Publications: Wiley-IEEE eBooks
IEEE Xplore Publications: IBM Periodicals
IEEE Xplore Publications: URSI Periodicals
IEEE Xplore Publications: VDE Conferences
IEEE Xplore Publications: Periodicals from China
IP.com Prior Art Database: The IP.com Journal
IP.com Prior Art Database: Internet Society RFC
IP.com Prior Art Database: IBM TDB Archive
IP.com Prior Art Database: Legacy Journals
IP.com Prior Art Database: Software Patent Institute
OnePetro.org: Periodicals at OnePetro.org
OnePetro.org: Conferences at OnePetro.org
Other Literature: IBM Redbooks
Other Literature: PubMed Central
Other Literature: arXiv.org

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L1	187807	(H04L29/08981 H04L67/34 H04L41/08-082).cpc.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 12:49 PM
L2	2173	distribut$3 adj2 (Software adj2 defin$3 adj2 Network (SDN)) d$1sdn	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 12:49 PM
L3	290	1 AND 2	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 12:49 PM
L4	16	3 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU,	OR	ON	ON	2022/04/07 12:49 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0355

			CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L5	2003388	(file application image container app) adj5 (portal merchant market$1place hub registry register$3 repository database DB directory)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 12:57 PM
L6	1517498	SDN VNF VNFCI Kubernetes Docker openflow openstack (MEC (Fog distributed mobile edge) adj5 (cloud computing))	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 12:58 PM
L7	78183	6 same2 5	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA,	OR	ON	ON	2022/04/07 12:58 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0356

			VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L8	3182	1 AND 7	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 12:58 PM
L9	962	8 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 12:58 PM
L10	726	8 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 12:59 PM
L18	3634458	SDN VNF VNFCI Kubernetes Docker openflow openstack MEC Fog cloud	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 01:04 PM
L19	86301	5 SAME 18	(US-PGPUB; USPAT;	OR	ON	ON	2022/04/07

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0357

			USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				01:05 PM
L20	3697	1 AND 19	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 01:05 PM
L21	687	20 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 01:05 PM
L22	492	20 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 01:06 PM
L23	161042	SDN VNF VNFCI Kubernetes Docker openflow openstack	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB,	OR	ON	ON	2022/04/07 01:11 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0358

			HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L24	88	21 AND 23	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 01:11 PM
L25	28912	23 SAME (version$3 updat$3 recent$3 up$1grad$3)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 01:16 PM
L27	4310	25 SAME (delay$3 paus$3 latency down$1time "without" interrupt$4)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT;	OR	ON	ON	2022/04/07 01:18 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0359

			IBM_TDB)				
L28	32	27 AND 1 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 01:18 PM
L29	940497	(version$3 updat$3 recent$3 up$1grad$3) adj20(delay$3 paus$3 latency down$1time "without" interrupt$4)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 01:27 PM
L30	172	21 AND 29	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 01:27 PM
L32	1	12/908606.app.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 01:32 PM
L33	2	("10893095").pn.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY,	OR	ON	ON	2022/04/07 05:35 PM

			CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L34	464641	(version$3 updat$3 recent$3 up$1grad$3) adj20(orchestrat$3 deploy$6 install$3 installation)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 05:42 PM
L35	740713	(version$3 updat$3 recent$3 up$1grad$3) adj10(hot delay$3 paus$3 latency down$1time "without" interrupt$4)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 05:43 PM
L36	20652	35 SAME 34	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH,	OR	ON	ON	2022/04/07 05:43 PM

			TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L38	2754564	SDN VNF VNFCI Kubernetes Docker openflow openstack cloud tenant multi$1tenant	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 05:44 PM
L39	16451	34 SAME 38	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 05:44 PM
L40	1327	36 AND 39	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/04/07 05:44 PM
L41	26	1 AND 40 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU,	OR	ON	ON	2022/04/07 05:45 PM

			CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N1	43002	SDN VNF VNFCI Kubernetes Docker openflow openstack.clm.	(US-PGPUB; USPAT)	OR	ON	ON	2022/04/07 01:14 PM
N2	1426678	(delay$3 paus$3 latency down$1time "without" interrupt$4).clm.	(US-PGPUB; USPAT)	OR	ON	ON	2022/04/07 01:19 PM
N3	645378	(version$3 updat$3 up$1grad$3).clm.	(US-PGPUB; USPAT)	OR	ON	ON	2022/04/07 01:19 PM
N4	35	N1 SAME N2 SAME N3	(US-PGPUB; USPAT)	OR	ON	ON	2022/04/07 01:19 PM
N6	3	N4 AND (@pd @ptad @prad @ad @rlad)<20130613	(US-PGPUB; USPAT)	OR	ON	ON	2022/04/07 01:19 PM

Web Search History

date, time	web site	search string
4/7/2022 1:41:22 PM	Google Scholar	"distributed SDN"
4/7/2022 1:42:07 PM	Google Scholar	updating distributed SDN application without substantial interruption
4/7/2022 1:42:23 PM	Google Scholar	updating distributed SDN application without substantial interruption [before:2013]
4/7/2022 1:43:40 PM	Google Scholar	[before:2013] updating distributed application in a cloud environment without substantial interruption
4/7/2022 1:45:04 PM	Google Scholar	[before:2013] updating multitenant software without substantial interruption
4/7/2022 1:45:11 PM	Google Scholar	[before:2013] updating multi tenant software without substantial interruption
4/7/2022 1:45:16 PM	Google Scholar	[before:2013] updating "multi tenant" software without substantial interruption

Google Scholar | updating distributed SDN application without substantial interruption | SIGN IN

Articles | About 3,460 results (0.03 sec) | My profile | My library

Any time
Since 2022
Since 2021
Since 2018
Custom range...
— 2013

Sort by relevance
Sort by date

Any type
Review articles

include patents
include citations

Create alert

Hotswap: Correct and efficient controller upgrades for software-defined networks [PDF] acm.org Full Text @ USPTO-STIC
L Vanbever, J Reich, T Benson, N Foster ... - ... on Hot topics in software ..., 2013 - dl.acm.org
... table entries and causes **substantial disruption** including losing ... , a system for **upgrading SDN** controllers in a **disruption**-free ... **without** fee provided that copies are **not** made or **distributed** ...
Save Cite Cited by 63 Related articles All 24 versions

Consistent updates for software-defined networks: Change you can believe in! [PDF] acm.org Full Text @ USPTO-STIC
M Reitblatt, N Foster, J Rexford, D Walker - Proceedings of the 10th ACM ..., 2011 - dl.acm.org
... term, is unlikely to lead to **significant** long-term progress. ... We believe **SDN** desperately needs such abstractions to make ... of adjusting link metrics to minimize **disruptions** [2, 3, 5]. The ...
Save Cite Cited by 266 Related articles All 18 versions

[PDF] **Walk the line: Consistent network updates with bandwidth guarantees** [PDF] acm.org
S Ghorbani, M Caesar - ... of the first workshop on Hot topics in software ..., 2012 - dl.acm.org
... Our algorithm runs in the **SDN** controller to orchestrate these ... The time interval [t1, t2) could be **significant** considering that ... to shield **applications** from **disruption** while migrating. ...
Save Cite Cited by 142 Related articles All 8 versions

B4: Experience with a globally-deployed software defined WAN [PDF] acm.org Full View
S Jain, A Kumar, S Mandal, J Ong ... - ACM SIGCOMM ..., 2013 - dl.acm.org
... scale B **outage**, pointing to challenges in both **SDN** and large... , and we can **upgrade** servers independently from the switch ... However, it has experienced one **substantial outage** that has ...
Save Cite Cited by 2661 Related articles All 60 versions

[PDF] **Onix: A distributed control platform for large-scale production networks** [PDF] usenix.org
T Koponen, M Casado, N Gude, J Stribling ... - ... USENIX Symposium on ..., 2010 - usenix.org
... network control as **Software**-Defined Networking (**SDN**). This is ... **disruption** to network traffic, the **application** may require the ... **distributed** coordination, race-conditions in state **updates** will ...
Save Cite Cited by 1932 Related articles All 64 versions

The road to SDN: An intellectual history of programmable networks [PDF] acm.org Full View
N Feamster, J Rexford, E Zegura - Queue, 2013 - dl.acm.org
... evolution of **SDN**, **substantially** predating **SDN** yet taking root ... load, minimizing transient **disruptions** during planned routing ... a firmware **upgrade**—vendors did **not** need to **upgrade** the ...
Save Cite Cited by 458 Related articles All 16 versions

OpenSession: SDN-based cross-layer multi-stream management protocol for 3D teleimmersion [PDF] ieee.org Full Text @ USPTO-STIC
A Arefin, R Rivas, R Tabassum... - 2013 21st IEEE ..., 2013 - ieeexplore.ieee.org
... Through **extensive** evaluation with multi-stream 3D ... : 1) modification of flow table does **not interrupt** on-going forwarding ... Below we formally present the **distributed** route **update** algorithm ...
Save Cite Cited by 19 Related articles All 10 versions

Abstractions for network update [PDF] acm.org

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0365

M Reitblatt, N Foster, J Rexford, C Schlesinger... - ACM SIGCOMM ..., 2012 - dl.acm.org

Full View

... correctly, and can result in **disruptions** such as transient outages, lost ... The emergence of **Software** Defined Networks (**SDN**) ... In this section, we describe optimizations that **substantially** ...

Save Cite Cited by 787 Related articles All 38 versions

[BOOK] **SDN: Software** Defined Networks: an authoritative review of network programmability technologies

TD Nadeau, K Gray - 2013 - books.google.com

... **updates** to this text going forward, given how young **SDN** still ... highly **distributed** world we live in today, in which **applications**... Incorporating a **significant** amount of example code from this ...

Save Cite Cited by 454 Related articles All 2 versions

Live migration of an entire network (and its hosts)

[PDF] acm.org

Full Text @ USPTO-STIC

E Keller, S Ghorbani, M Caesar, J Rexford - Proceedings of the 11th ..., 2012 - dl.acm.org

... After a brief overview of **SDN** and the state in the switches, ... with the switches to **update** the rules and handle events: ... **substantially** reduced **outage** time (measured by packet loss), ...

Save Cite Cited by 142 Related articles All 27 versions

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0366

PTO/SB/30 (11-17)
Approved for use through 11/30/2020. 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

Address to:
Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Application Number	17/142,983
Filing Date	1/6/2021
First Named Inventor	Taaghol
Art Unit	
Examiner Name	
Attorney Docket Number	TAAG001CONT3

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. See Instruction Sheet for RCEs (not to be submitted to the USPTO) on page 2.

1. 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).

   a. ☐ 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.

      i. ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____

      ii. ☐ Other _____

   b. ☑ Enclosed

      i. ☑ Amendment/Reply

      ii. ☐ Affidavit(s)/ Declaration(s)

      iii. ☐ Information Disclosure Statement (IDS)

      iv. ☐ Other _____

2. Miscellaneous

   a. ☐ 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)

   b. ☐ Other _____

3. Fees The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.

   a. ☐ The Director is hereby authorized to charge the following fees, any underpayment of fees, or credit any overpayments, to Deposit Account No. _____.

      i. ☐ RCE fee required under 37 CFR 1.17(e)

      ii. ☐ Extension of time fee (37 CFR 1.136 and 1.17)

      iii. ☐ Other _____

   b. ☐ Check in the amount of $ _____ enclosed

   c. ☐ Payment by credit card (Form PTO-2038 enclosed)

   d. ■ Payment by EFS-Web

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

Signature	/Christopher J. Horgan/	Date	July 19, 2022
Name (Print/Type)	Christopher J. Horgan	Registration No.	40,394

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being EFS-Web transmitted to the United States Patent and Trademark Office (USPTO), deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop RCE, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450 or facsimile transmitted to the USPTO on the date shown below.

Signature			
Name (Print/Type)		Date	

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. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Mail Stop RCE, 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.

Electronic Patent Application Fee Transmittal

Application Number:	17142983
Filing Date:	06-Jan-2021
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Filer:	Christopher James Horgan
Attorney Docket Number:	TAAG001CONT3

Filed as Small Entity

Filing Fees for Utility under 35 USC 111(a)

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:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension - 1 month with $0 paid	2251	1	110	110
Miscellaneous:				
RCE- 1ST REQUEST	2801	1	680	680
		Total in USD ($)		**790**

Electronic Acknowledgement Receipt

EFS ID:	46267413
Application Number:	17142983
International Application Number:	
Confirmation Number:	2943
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Customer Number:	112449
Filer:	Christopher James Horgan
Filer Authorized By:	
Attorney Docket Number:	TAAG001CONT3
Receipt Date:	25-JUL-2022
Filing Date:	06-JAN-2021
Time Stamp:	20:00:34
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$790
RAM confirmation Number	E20227OK02063353
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Continued Examination (RCE)	TAAG001CONT3_RCE_____.pdf	169822 0e57507d70ae53e8d9a774d271e872aaeecc3ef4	no	1
Warnings:					
This is not a USPTO supplied RCE SB30 form.					
Information:					
2	Amendment Submitted/Entered with Filing of Continued Prosecution Application (CPA)/Request for Continued Examination(RCE)	TAAG001CONT3_RESPONSE_____.pdf	62108 c0bb4d78245da630d85881d156093b747af798cb	no	8
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	40603 3f3a1bf6758b310896762661d51e0f0a1583e658	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	272533		

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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Taaghol
Serial No.: 17/142,983
Filed: 1/6/2021
Attorney Docket No.: TAAG001CONT3

Examiner: Higa
Art Unit: 2441

For: DISTRIBUTED SOFTWARE DEFINED NETWORKING

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

Dear Sir:

RESPONSE

Applicants offer the following election in response to the Office Action (hereinafter "Office Action") mailed on April 8, 2022.

Amendments to the Claims begin on page 2 of this paper.
Remarks begin on page 7 of this paper.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0372

In The Claims:

1. (Canceled)
2. (Canceled)
3. (Canceled)
4. (Canceled)
5. (Canceled)
6. (Canceled)
7. (Canceled)
8. (Canceled)
9. (Canceled)
10. (Canceled)
11. (Canceled)
12. (Canceled)
13. (Canceled)
14. (Canceled)
15. (Canceled)
16. (Canceled)
17. (Canceled)
18. (Canceled)

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0373

Please add the following new claims:

19. (New) A system comprising a plurality of network elements, said network elements comprising:

a programmable network device adapted to host a plurality of first network applications;

a programmable cloud device adapted to host a plurality of second network applications, wherein the plurality of first network applications in the programmable network device and the plurality of second network applications in the programmable cloud device are in secure communication with each other to form distributed applications; and

wherein the programmable network device and programmable cloud device form unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) that allow for simultaneous programming of the programmable network device and programmable cloud device independent of network device hardware and cloud device hardware.

20. (New) The system of claim 19, wherein each of the programmable network device and programmable cloud device include an access network interface unit configured to send and receive communications and a processor with a memory associated with the access network interface unit.

21. (New) The system of claim 19 further comprising:

an application management portal capable of managing life cycles of the distributed applications; and

an infrastructure application market place in communication with the application management portal, said infrastructure market place capable of providing tested and certified distributed applications to the application management portal.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0374

22. (New) The system of claim 21, wherein the application management portal is capable of at least one of the group consisting of: receiving new applications from the infrastructure application market place; test said distributed applications prior to deployment; provision said distributed applications; and deprovision said distributed applications.

23. (New) The system of claim 19, wherein the at least one of the first network applications is responsible for a plurality of functional capabilities of the programmable network device and is capable of at least one of the group consisting of: updating firmware of the programmable network device; updating default communication protocols of the programmable network device; initializing programmable network device settings; boot loading the programmable network device; providing security digital certificates; providing operating system (OS) and kernel upgrades; and upgrading and commissioning agents for the first network application upgrades and related procedures.

24. (New) The system of claim 19, wherein the at least one of the second network applications is responsible for a plurality of functional capabilities of the programmable cloud device and is capable of at least one of the group consisting of: upgrading and commissioning agents for the second network application updates and related procedures; hypervisor upgrades; virtual machine and isolated execution environment lifecycle management; security digital certificate management; OS and kernel upgrades; and default communication protocols.

25. (New) The system of claim 19, wherein the progrmmable network device is an LTE eNB and the programmable cloud device hosts an LTE core network functions in separate virtual machines from the group consisting of: MME, S-GW, and P-GW.

26. (New) The system of claim 19, wherein the programmable network device and programmable cloud device further comprise:

a virtual fabric which provides a secure communication layer for said at least one of the first network applications and said at least one of the second network

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0375

applications.

27. (New) The system of claim 19, further comprising:

a second programmable network device which includes at least one of the first network applications whichis the same as the at least one of the first network applications in the first programmable network device and where each of the first network applications can communicate directly.

28. (New) The system of claim 21, wherein the programmable network device and the programmable cloud device further comprise:

a distributed resource service (DRS) which controls access to a plurality of resources upon receiving instructions from the application management portal.

29. (New) The system of claim 28, wherein the resources include at least one of a group consisting of: firewall; data and statistics; storage; compute; load evaluator; general settings; routing engine; virtual machine master controller; power controller; and wireless engine.

30. (New) The system of claim 19, wherein the programmable network device and programmable cloud device further comprise:

a distributed resource service which is capable of providing at least one service from a group consisting of: exposing application programming interfaces (APIs) to other applications; configuring and managing platform resources; policy enforcement and authorization of applications access to platform resource and other application's APIs; and policy conflict resolution.

31. (New) The system of claim 19, wherein the programmable network device and the programmable cloud device further comprise:

a distributed content service wherein contents generated by the at least one of the plurality of first network applications are shared with the at least one of the plurality of second network applications.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0376

32. (New) The system of claim 21, wherein the flexible network device and flexible cloudplatform further comprise:

a distributed notification service wherein notice of a predetermined event is sent from the at least one fxDeviceApp to the at least one fxCloudApp.

33. (New) The system of claim 28, wherein the DRS further includes a load controller; and wherein the load controller is adapted to monitor loads on the at least one of the plurality of first network applications and at least one of the plurality of second network applications and effect change in accordance with thresholds received from the application management portal.

34. (New) The system of claim 19, wherein the programmable network device, at least one of the plurality of first network applications, the programmable cloud device, and the at least one of the plurality of second network applications each have unique identities.

35. (New) The system of claim 19, wherein the at least one of the plurality of first network applications and at least one of the plurality of second network applications are given an access level to determine which APIs can be accessed.

36. (New) The system of claim 19, wherein the at least one programmable network device and programmable cloud device are adapted to verify the authenticity and integrity of updates.

37. (New) A system comprising a plurality of network elements, said network elements comprising:

a programmable cloud device having a plurality of virtual machines, wherein of the virtual machines has at least one of a plurality of first network applications; and

a programmable network device adapted to process data flows having at least one of a plurality of second network applications, wherein the first and second network applications are in secure communication to form a distributed application.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0377

38. (New) A system comprising a plurality of network elements, said network elements comprising:

a programmable cloud device having a plurality of zones, wherein of the plurality of zones has at least one of a plurality of first network applications; and

a programmable network device adapted to process data flows having at least one of a plurality of second network applications, wherein the first and second network applications are in secure communication to form a distributed application.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0378

REMARKS

Claims 1-18 are canceled. Claims 19-38 have been added.

DOUBLE PATENTING REJECTIONS

Claims 1-18 are rejected on the ground of statutory double patenting as being unpatentable over U.S. Patent No. 10,893,095.

Applicant has canceled claims 1-18 and added new claims 19-38 which are not coextensive in scope with the prior applications of this family. No new matter has been added in the new claims and/or limitations presented in this Amendment.

Respectfully submitted,
ROARK IP
By: /Christopher J. Horgan/

Christopher J. Horgan
Registration No. 40,394
1438 Dahlia Loop
San Jose, CA 95126
Telephone: (408) 687-8306

Date: July 19, 2022

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0379

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 17/142,983	Filing Date 01/06/2021	☐ 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 $50 =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*		x $240 =	
☐ 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	07/25/2022	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 20	Minus	** 20	= 0		x $50 =	0
	Independent (37 CFR 1.16(h))	* 3	Minus	*** 3	= 0		x $240 =	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 $0 =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=		x $0 =	
	☐ 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.	LIE
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".	/DORIS M KING/
*** 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.	

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.

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 17/142,983	Filing Date 01/06/2021	☐ 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 $50 =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*		x $240 =	
☐ 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	07/25/2022	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 20	Minus	** 20	= 0		x $50 =	0
	Independent (37 CFR 1.16(h))	* 3	Minus	*** 3	= 0		x $240 =	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 $0 =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=		x $0 =	
	☐ 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.	LIE
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".	/DORIS M KING/
*** 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.	

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.

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.
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3	2943

112449 7590 09/13/2022

Roark IP
1438 Dahlia Loop
San Jose, CA 95126

EXAMINER	
HIGA, BRENDAN Y	

ART UNIT	PAPER NUMBER
2441	

MAIL DATE	DELIVERY MODE
09/13/2022	PAPER

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.

Office Action Summary	Application No. 17/142,983	Applicant(s) Taaghol et al.	
	Examiner BRENDAN HIGA	Art Unit 2441	AIA (FITF) Status Yes

-- 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 07/25/2022.

☐ 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) 19-38 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) 19-38 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 01/06/2021 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

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on July 25, 2022 has been entered.

Claims 19-38 are pending.

Response to Arguments

Applicant's new claims 19-38 overcome the previous statutory Double Patenting rejection.

Claim Rejections - 35 USC § 112

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

> (a) IN GENERAL.—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 or joint inventor of carrying out the invention.

The following is a quotation of the first paragraph of pre-AIA 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.

Claims 19-36 are rejected under 35 U.S.C. 112(a) or 35 U.S.C. 112 (pre-AIA), 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 or a joint inventor, or for applications subject to pre-AIA 35 U.S.C. 112, the inventor(s), at the time the application was filed, had possession of the claimed invention.

As per claims 19-36 the limitation "…wherein the programmable network device and programmable cloud device *form unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) that allow for simultaneous programming of the programmable network device and programmable cloud device independent of network device hardware and cloud device hardware*", does not appear to be described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor or a joint inventor, or for applications subject to pre-AIA 35 U.S.C. 112, the inventor(s), at the time the application was filed, had possession of the claimed invention.

Particularly Applicant's specification is completely silent as to the steps or processes for the "*simultaneous programming of the programmable network device and programmable cloud device independent of network device hardware and cloud device hardware".*

Claim Rejections - 35 USC § 102

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

> A person shall be entitled to a patent unless –
>
> (a)(1) the claimed invention was patented, described in a printed publication, or in public use, on sale or otherwise available to the public before the effective filing date of the claimed invention.
>
> (a)(2) the claimed invention was described in a patent issued under section 151, or in an application for patent published or deemed published under section 122(b), in which the patent or application, as the case may be, names another inventor and was effectively filed before the effective filing date of the claimed invention.

Claims 37-38 are rejected under 35 U.S.C. 102(a)(2) as being anticipated by Van Der Merwe et al. (US 2013/0054763) ("Van der Merwe").

As per claim 37, Van Der Merwe, teaches a system comprising a plurality of network elements (*see Fig. 1-2*), said network elements comprising:

a programmable cloud device having a plurality of virtual machines (*i.e., a cloud computing data center comprising virtual machines, see ¶0025, and ¶0083, i.e., "virtualized content server 510 includes the virtual machines VM X3 and VM X4 provisioned for the Client X"*), wherein each of the virtual machines has at least one of a plurality of first network applications (*hosting enterprise applications, see ¶0083*); and

a programmable network device (*i.e., edge router, see abstract and Fig. 2, ref. 232*) adapted to process data flows (*i.e., network traffic, including filtering and/or firewall functioning, read as "processing" data flows, see ¶0063*) having at least one of a plurality of second network applications (*i.e., operating as a logical router, see ¶0062, read as a second network application*), wherein the first and second network

applications are in secure communication to form a distributed application (*i.e., "virtualized platform", see ¶0002, and/or "an overlay network", see ¶0019, read as a distributed application, also see ¶0019, "an overlay network to provide secure connection(s)", also see ¶0105, "The updating of the VRF tables causes the routers 204-208 and 232 to transmit securely communications received from the interface 114 of a VPMN with a destination address of a virtual machine through the IP network 102 along the virtual route 326")*.

As per claim 38, Van der Merwe teaches a system comprising a plurality of network elements (*see Fig. 1-2*), said network elements comprising:

a programmable cloud device having a plurality of zones (*i.e., a cloud computing data center comprising virtual machines, see ¶0025, read as isolated "zones")*, wherein each of the plurality of zones has at least one of a plurality of first network applications (*hosting enterprise applications, see ¶0083)*; and

a programmable network device adapted to process data flows (*i.e., network traffic, including filtering and/or firewall functioning, read as "processing" data flows, see ¶0063*) having at least one of a plurality of second network applications, wherein the first and second network applications are in secure communication to form a distributed application (*i.e., "virtualized platform", see ¶0002, and/or "an overlay network", see ¶0019, read as a distributed application, also see ¶0019, "an overlay network to provide secure connection(s)", also see ¶0105, "The updating of the VRF tables causes the routers 204-208 and 232 to transmit securely communications received from the*

interface 114 of a VPMN with a destination address of a virtual machine through the IP network 102 along the virtual route 326").

Allowable Subject Matter

Claims 19-36 are allowed over the prior art.

The following is an examiner's statement of reasons for allowance:

The prior art does not teach or render obvious to one of ordinary skill in the art, before the earliest effective filing date of the claimed invention, in the specific combinations and manner recited within the claims, the features of:

"…wherein the programmable network device and programmable cloud device *form unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) that allow for simultaneous programming of the programmable network device and programmable cloud device independent of network device hardware and cloud device hardware*."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to BRENDAN HIGA whose telephone number is (571)272-5823. The examiner can normally be reached on Monday - Friday 8:30 AM - 5:00 PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an

interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at http://www.uspto.gov/interviewpractice.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wing Chan can be reached on (571) 272-7493. 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). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/BRENDAN Y HIGA/
Primary Examiner, Art Unit 2441

Notice of References Cited	Application/Control No. 17/142,983	Applicant(s)/Patent Under Reexamination Taaghol et al.	
	Examiner BRENDAN HIGA	Art Unit 2441	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-8762501-B2	06-2014	Kempf; James	H04L47/825	709/228
*	B	US-20150071053-A1	03-2015	Kempf; James	H04W24/02	370/216
*	C	US-20130054763-A1	02-2013	Van der Merwe; Jacobus	H04W8/12	709/220
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC 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
	17/142,983	Taaghol et al.
	Examiner	Art Unit
	BRENDAN HIGA	2441

CPC - Searched*		
Symbol	Date	Examiner
H04L41/08,0895,22; 45/00,56-563,586,64,645,655,76; 67/34	09/05/2022	BH

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
see SEARCH history	09/05/2022	BH
IP.com (NPL+Patents)	09/05/2022	BH
Google Scholar (see search history and results)	09/05/2022	BH
Reviewed prosecution history of US application 14/295331, particularly final rejection mailed on 12/28/2016	09/02/2022	BH

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
H04L	41/08,0895,22; 45/00,56-563,586,64,645,655,76; 67/34	09/05/2022	BH

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0391

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	**Art Unit**
	BRENDAN HIGA	2441

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

CLAIMS

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

CLAIM		DATE								
Final	Original	01/01/2022	04/07/2022	09/04/2022						
	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			✓						
	37			✓						
	38			✓						

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0392

IQSearchText-202209051736

Main Concept Document
==========
US9843624
Title: Distributed software defined networking
Abstract: The Distributed Software Defined Network (dSDN) disclosed herein is an end-to-end architecture that enables secure and flexible programmability across a network with full lifecycle management of services and infrastructure applications (fxDeviceApp). The dSDN also harmonizes application deployment across the network independent of the hardware vendor. As a result, the dSDN simplifies the network deployment lifecycle from concept to design to implementation to decommissioning.

(No Concept Modifiers)
==========

Filters (1)
==========
Publication Date: To 2013-12-31

Options
==========
De-dup: Extended Family Numbers (PAT only)
Cut-off: None
Sort: Relevance

Content: Patents + NPL (190)
==========
US Patents
US Designs
US Applications
EPO Patents
EPO Applications
China Patents
China Applications
Japan Patents
Japan Applications
Korea Patents
Korea Applications
WIPO Applications
Argentina Patents
Argentina Applications
Brazil Patents
Brazil Applications
Canada Patents

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0393

Canada Applications
Chile Patents
Chile Applications
Colombia Applications
Costa Rica Applications
Cuba Patents
Cuba Applications
Dominican Republic Applications
Ecuador Patents
Ecuador Applications
El Salvador Applications
Guatemala Applications
Honduras Applications
Mexico Patents
Mexico Applications
Nicaragua Patents
Panama Applications
Peru Applications
Trinidad & Tobago Patents
Uruguay Applications
Austria Patents
Austria Applications
Belarus Patents
Belgium Patents
Belgium Applications
Bosnia & Herzegovina Patents
Bosnia & Herzegovina Applications
Bulgaria Patents
Bulgaria Applications
Croatia Patents
Croatia Applications
Czech Republic Patents
Czech Republic Applications
Czechoslovakia Patents
Czechoslovakia Applications
Denmark Patents
Denmark Applications
Estonia Patents
Estonia Applications
EUIPO Patents
Finland Patents
Finland Applications
France Patents
France Applications
Germany Patents
Germany Applications
Great Britain Patents
Great Britain Applications

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0394

Greece Patents
Greece Applications
Hungary Patents
Hungary Applications
Iceland Patents
Iceland Applications
Ireland Patents
Ireland Applications
Italy Patents
Italy Applications
Latvia Patents
Latvia Applications
Lithuania Patents
Lithuania Applications
Luxembourg Patents
Malta Patents
Monaco Patents
Montenegro Patents
Montenegro Applications
Netherlands Patents
Netherlands Applications
Norway Patents
Norway Applications
Poland Patents
Poland Applications
Portugal Patents
Portugal Applications
Republic of Moldova Patents
Republic of Moldova Applications
Romania Patents
Romania Applications
San Marino Patents
San Marino Applications
Serbia Patents
Serbia Applications
Slovakia Patents
Slovakia Applications
Slovenia Patents
Spain Patents
Spain Applications
Sweden Patents
Sweden Applications
Switzerland Patents
Switzerland Applications
Ukraine Patents
Yugoslavia/Serbia and Montenegro Patents
Yugoslavia/Serbia and Montenegro Applications
Armenia Patents

Australia Patents
Australia Applications
Cyprus Patents
Gulf Cooperation Council Patents
Hong Kong Patents
India Patents
India Applications
Indonesia Patents
Indonesia Applications
Israel Applications
Jordan Patents
Jordan Applications
Kyrgyzstan Patents
Macao Applications
Malaysia Patents
Mongolia Patents
New Zealand Patents
Philippines Patents
Philippines Applications
Saudi Arabia Patents
Saudi Arabia Applications
Singapore Patents
Singapore Applications
Taiwan Patents
Taiwan Applications
Tajikistan Patents
Tajikistan Applications
Thailand Patents
Thailand Applications
Uzbekistan Patents
Vietnam Patents
Algeria Patents
ARIPO Patents
ARIPO Applications
Egypt Patents
Kenya Patents
Malawi Patents
Morocco Patents
Morocco Applications
OAPI Patents
South Africa Patents
Tunisia Applications
Zambia Patents
Zimbabwe Patents
EAPO Patents
EAPO Applications
Georgia Patents
Georgia Applications

Kazakhstan Patents
Kazakhstan Applications
Russia Patents
Russia Applications
Turkey Patents
Turkey Applications
IEEE Xplore Publications: IEEE Periodicals
IEEE Xplore Publications: IEEE Conferences
IEEE Xplore Publications: IEEE Standards
IEEE Xplore Publications: IEEE Early Access
IEEE Xplore Publications: SMPTE Periodicals
IEEE Xplore Publications: SMPTE Conferences
IEEE Xplore Publications: SMPTE Standards
IEEE Xplore Publications: MIT Press eBooks
IEEE Xplore Publications: Wiley-IEEE eBooks
IEEE Xplore Publications: IBM Periodicals
IEEE Xplore Publications: URSI Periodicals
IEEE Xplore Publications: VDE Conferences
IEEE Xplore Publications: Periodicals from China
IP.com Prior Art Database: The IP.com Journal
IP.com Prior Art Database: Internet Society RFC
IP.com Prior Art Database: IBM TDB Archive
IP.com Prior Art Database: Legacy Journals
IP.com Prior Art Database: Software Patent Institute
OnePetro.org: Periodicals at OnePetro.org
OnePetro.org: Conferences at OnePetro.org
Other Literature: IBM Redbooks
Other Literature: PubMed Central
Other Literature: arXiv.org

Mobile **fog**: A **programming** model for large-scale applications on the **internet** of things

K Hong, D Lillethun, U Ramachandran... - ... on Mobile **cloud** ..., 2013 - dl.acm.org

... On the other hand, we provide the point-to-point **API** to support communication with mobile ... events stored at each Mobile **Fog** process running on **edge fog** nodes in the MCEP scenario. ...

☆ Save 99 Cite Cited by 646 Related articles All 9 versions »

Methods, devices, and media for securely utilizing a non-secured, **distributed**, virtualized **network** resource with applications to **cloud**-computing security and ...

G Parann-Nissany - US Patent App. 12/887,547, 2011 - Google Patents

... this **API** appears similar to the **API** of the **cloud** resource (eg HTTP-based REST **API**), differing from the **cloud**-resource **API** only ... of URLs and addresses by the **cloud** interface proxy. The ...

☆ Save 99 Cite Cited by 88 Related articles All 2 versions »

Clouds of virtual machines in **edge networks**

A Manzalini, R Minerva, F Callegati... - IEEE ..., 2013 - ieeexplore.ieee.org

... This floating "**fog**" of ICT resources at the **edge** will give rise to new business models based on ... an open application **programming** interface (**API**) for **programming** the **network** at various ...

☆ Save 99 Cite Cited by 116 Related articles All 2 versions

Cloud computing gateway, **cloud** computing hypervisor, and methods for implementing same

DC Hilt - US Patent 8,514,868, 2013 - Google Patents

... account 960, and the Sungard **cloud** provider **network** account 965) has a unique **API** and a corresponding back-end plug-in 915. It will be appreciated that, in other embodiments, each ...

☆ Save 99 Cite Cited by 425 Related articles All 4 versions »

Bridging enterprise **networks** into **cloud**

P Lahiri, PK Patel, DA Maltz, A Greenberg... - US Patent App. 12 ..., 2010 - Google Patents

... In another implementation, management service 120 may provide an application **programming** interface (**API**) of management functions to users. Management service 120 may be ...

☆ Save 99 Cite Cited by 291 Related articles All 2 versions »

Application Aware **Networking**

WC Stillwell III, PE Harris - US Patent App. 11/971,355, 2009 - Google Patents

... , that cutting **edge** content always exceeds the capacity of the **network** to deliver it. For ... **programming** interface (**API**), for use by a developer or tester, and/or included within the **network** ...

☆ Save 99 Cite Cited by 64 Related articles All 2 versions »

Java application framework for use in a content delivery **network** (CDN)

AT Davis, J Parikh, S Pichai, E Ruvinsky ... - US Patent ..., 2006 - Google Patents

... a globally **distributed** computing platform, such as an **Internet** content delivery **network** (CDN). ... **distributed edge** layer and a centralized origin layer. In a representative embodiment, the ...

☆ Save 99 Cite Cited by 588 Related articles All 4 versions »

Applying policies to **schedule network** bandwidth among virtual machines

Y Zuo, HY Chau, HH Vo, SN Arafeh... - US Patent ..., 2013 - Google Patents

... In one instance, updates to the client contracts may be made via a user-mode **API** that ... absolute bandwidth may be entered at the user-mode **API**. The "relative weighting" and the "...

☆ Save 99 Cite Cited by 158 Related articles All 4 versions »

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0398

Application Resource Manager over a **Cloud**

MV Devarakonda, NA Joukov, B Pfitzmann... - US Patent App. 13 ..., 2013 - Google Patents

... a communications **network**. In a **distributed cloud** computing environment, **program** modules ... here that either this also contains an **API** (application **programming** interface), web service ...

☆ Save 99 Cite Cited by 122 Related articles All 2 versions »

System and method for grid and **cloud** computing

R Buyya, S Venugopal, X Chu, K Nadiminti - US Patent 8,230,070, 2012 - Google Patents

... **Network** 10 allows a user to implement various parallel and **distributed programming** models, as is ... In addition, the threading **programming** model, and core MPI **APIs** or the Map Reduce ...

☆ Save 99 Cite Cited by 178 Related articles All 4 versions »

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0399

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L1	3	("20140241158" "20120300615" "20090276771").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/09/02 10:48 AM
L2	1	14/295331.app.	(US-PGPUB; USPAT)	OR	ON	ON	2022/09/02 12:10 PM
L3	92	("7114021" | "7630706" | "8132166" | "8528108" | "8650299" | "8756595" | "8843933" | "9226333" | "20020016954" | "20040107417" | "20040237081" | "20050246703" | "20070078988" | "20080209491" | "20090007091" | "20090113327" | "20090276771" | "20090300057" | "20110075611" | "20120300615" | "20130007254" | "20130160147" | "20130263104" | "20140052976" | "20140241158" | "20140254373" | "20140259012" | "20140317293" | "20150173121").pn. OR ("9843624").urpn. AND (PGPB | USPT | USOC).dbnm.	(US-PGPUB; USPAT; USOCR)	OR	ON	ON	2022/09/02 12:43 PM
L4	3	("20150071053" "8762501" "20130054763").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/09/02 04:02 PM
L5	78988	(H04L45/00,586,64,645,655,76).cpc.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/04 01:05 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0400

L6	6574	(docker kubernetes SDN NFV Open$1flow open$1stack VM virtual$4) SAME (simultaneous$2 concurrent$3 synchronous$3) adj5 (program$5 install$6 deploy$5 orchestrat$4)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/04 01:06 PM
L7	14	5 AND 6 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/04 01:07 PM
L8	65635	(H04L41/08,0895,22).CPC.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/04 03:34 PM
L11	11	6 AND 8 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL,	OR	ON	ON	2022/09/04 03:34 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0401

			NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L12	1678	(cloud) SAME (simultaneous$2 concurrent$3 synchronous$3) adj5 (program$5 install$6 deploy$5 orchestrat$4)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/04 03:36 PM
L13	0	12 AND (5 8) AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/04 03:36 PM
L14	1	("10893095").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/09/05 12:48 PM
L15	39	(Artificial Intelligence) More like doc: US-10893095-B1 Text: (US-9843624-B1 OR US-10686871-B1 OR US-20180139109-A1 OR US-20220038530-A1 OR US-10904108-B2 OR US-20200244552-A1 OR US-20200067797-A1 OR US-11153367-B2 OR US-10523531-B2 OR CN-114095196-A	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/05 12:48 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0402

		OR US-10911558-B1 OR US-10887130-B2 OR CN-113039768-A OR US-10904107-B2 OR WO-2021080800-A1 OR US-10819650-B2 OR US-20210119971-A1 OR CN-114365454-A OR US-11019035-B2 OR US-20200412823-A1).did.					
L16	40	("7114021" | "7630706" | "8132166" | "8495611" | "8528108" | "8650299" | "8756595" | "8843933" | "9226333" | "20020016954" | "20040107417" | "20040237081" | "20050246703" | "20060271341" | "20070078988" | "20080209491" | "20090007091" | "20090113327" | "20090276771" | "20090300057" | "20110075611" | "20110265077" | "20120066665" | "20120096071" | "20120266156" | "20120300615" | "20130007254" | "20130160147" | "20130223442" | "20130263104" | "20140052976" | "20140241158" | "20140254373" | "20140259012" | "20140317293" | "20150160877" | "20150173121" | "20150242237" | "20160224460" | "20190165997").pn. OR ("10893095").urpn. AND (PGPB | USPT | USOC).dbnm.	(US-PGPUB; USPAT; USOCR)	OR	ON	ON	2022/09/05 12:49 PM
L17	0	(6 12) AND (15 16)	(US-PGPUB; USPAT; USOCR)	OR	ON	ON	2022/09/05 12:49 PM
L18	2	5 AND 8 AND (15 16)	(US-PGPUB; USPAT; USOCR)	OR	ON	ON	2022/09/05 12:50 PM
L19	0	17/142983.app.	(US-PGPUB; USPAT;	OR	ON	ON	2022/09/05

			USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)				02:35 PM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N1	175	(cloud docker kubernetes SDN NFV Open$1flow open$1stack VM virtual$4) SAME (simultaneous$2 concurrent$3 synchronous$3) adj5 (program$5 install$6 deploy$5 orchestrat$4).clm.	(US-PGPUB; USPAT)	OR	ON	ON	2022/09/04 03:37 PM
N2	126	N1 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT)	OR	ON	ON	2022/09/04 03:37 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0404

Web Search History

date, time	web site	search string
9/5/2022 1:30:47 PM	Google Scholar	[before:2013] programmable cloud or fog or edge for wireless network
9/5/2022 1:31:00 PM	Google Scholar	[before:2013] programmable cloud or fog or edge for wireless network and "simulatneous"
9/5/2022 1:31:03 PM	Google Scholar	[before:2013] programmable cloud or fog or edge for wireless network and "simultaneous"
9/5/2022 1:31:49 PM	Google Scholar	programmable cloud or fog or edge for wireless network and "simultaneous" [before:2013]
9/5/2022 1:55:27 PM	Google Scholar	[before:2013] API for programming "distributed" cloud or fog or edge network
9/5/2022 1:55:36 PM	Google Scholar	API for programming "distributed" cloud or fog or edge network [before:2013]
9/5/2022 2:00:41 PM	Google Scholar	[before:2013] API for programming "distributed" "OpenFlow" applications
9/5/2022 2:01:35 PM	Google Scholar	[before:2013] API for "simultaneous" programming "distributed" cloud or SDN network
9/5/2022 2:02:00 PM	Google Scholar	[before:2013] API for programming "distributed" cloud or SDN network

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0405

Electronic Acknowledgement Receipt

EFS ID:	46648801
Application Number:	17142983
International Application Number:	
Confirmation Number:	2943
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Customer Number:	112449
Filer:	Christopher James Horgan
Filer Authorized By:	
Attorney Docket Number:	TAAG001CONT3
Receipt Date:	19-SEP-2022
Filing Date:	06-JAN-2021
Time Stamp:	22:41:39
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/Request for Reconsideration-After Non-Final Rejection	TAAG001CONT3_RESPONSE_____.pdf	92731 58c844be5813c5097b0f89b7f0ac13ccdd52db22	no	9

Warnings:

Information:	
Total Files Size (in bytes):	92731

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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Taaghol
Serial No.: 17/142,983
Filed: 1/6/2021
Attorney Docket No.: TAAG001CONT3

Examiner: Higa
Art Unit: 2441

For: DISTRIBUTED SOFTWARE DEFINED NETWORKING

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

Dear Sir:

RESPONSE

Applicants offer the following election in response to the Office Action (hereinafter "Office Action") mailed on September 13, 2022.

Amendments to the Claims begin on page 2 of this paper.
Remarks begin on page 8 of this paper.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0408

In The Claims:

1. (Canceled)
2. (Canceled)
3. (Canceled)
4. (Canceled)
5. (Canceled)
6. (Canceled)
7. (Canceled)
8. (Canceled)
9. (Canceled)
10. (Canceled)
11. (Canceled)
12. (Canceled)
13. (Canceled)
14. (Canceled)
15. (Canceled)
16. (Canceled)
17. (Canceled)
18. (Canceled)

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0409

19. (Currently Amended) A system comprising ~~a plurality of network elements, said network elements comprising~~:

a programmable network device adapted to host a plurality of [first] network device applications;

a programmable cloud device adapted to host a plurality of [second network] cloud applications, wherein the plurality of [first] network device applications ~~in the programmable network device~~ and the plurality of [second network] cloud applications ~~in the programmable cloud device~~ are in secure communication with each other to form distributed applications; and

wherein the ~~programmable~~ plurality of network device applications and ~~programmable~~ plurality of cloud applications ~~device~~ form unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) [that] to ~~allow for simultaneous programming of~~ simultaneously program the [programmable] plurality of network device applications and [programmable] plurality of cloud applications [device] independent of network device hardware and cloud device hardware.

20. (Currently Amended) The system of claim 19, wherein ~~each of~~ the programmable network device and the programmable cloud device each include an access network interface unit configured to send and receive communications and a processor with a memory associated with the access network interface unit.

21. (Currently Amended) The system of claim 19 further comprising:

an application management portal capable of managing life cycles of the distributed applications; and

an infrastructure application ~~market place~~ marketplace in communication with the applicationmanagement portal, said infrastructure ~~market place~~ marketplace capable of providing tested ~~and certified~~ distributed applications to the application management portal.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0410

22. (Currently Amended) The system of claim 21, wherein the application management portal is capable of at least one of the group consisting of: receiving new applications from the infrastructure application marketplace ~~market place~~; testing said distributed applications prior to deployment; provision said distributed applications; and deprovision said distributed applications.

23. (Currently Amended) The system of claim 19, wherein [the] at least one of the ~~first~~ plurality of network device applications is responsible for a plurality of functional capabilities of the programmable network device and is capable of at least one of the group consisting of: updating firmware of the programmable network device; updating default communication protocols of the programmable network device; initializing programmable network device settings; boot loading the programmable network device; providing security digital certificates; providing operating system (OS) and kernel upgrades; and upgrading and commissioning agents for [the first] at least one of the plurality of network device ~~application~~ applications upgrades and related procedures.

24. (Currently Amended) The system of claim 19, wherein [the] at least one of the ~~second network~~ plurality of cloud applications is responsible for a plurality of functional capabilities of the programmable cloud device and is capable of at least one of the group consisting of: upgrading and commissioning agents for the [second network] plurality of cloud application updates ~~and related procedures~~; hypervisor upgrades; virtual machine and isolated execution environment lifecycle management; security digital certificate management; operating system (OS) and kernel upgrades; and default communication protocols.

25. (Currently Amended) The system of claim 19, wherein the ~~progrmmable~~ programmable network device is an LTE eNB and the programmable cloud device hosts [an] LTE core network functions in separate virtual machines from the group consisting of: MME, S-GW, and P-GW.

26. (Currently Amended) The system of claim 19, ~~wherein the programmable network device and programmable cloud device further comprise~~ further comprising:

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0411

a virtual fabric which provides a secure communication layer for said at least one of the [first] plurality of network device applications and said at least one of the [second network] plurality of cloud applications.

27. (Currently Amended) The system of claim 19, further comprising:

a second programmable network device which includes at least one of the [first] plurality of network device applications which is the same as [the] at least one of the [first] plurality of network device applications in the [first] programmable network device and where each of the [first] network device applications can communicate directly.

28. (Currently Amended) The system of claim 21~~, wherein the programmable network device and the programmable cloud device further comprise~~ further comprising:

a distributed resource service (DRS) which controls access to a plurality of resources upon receiving instructions from the application management portal.

29. (Currently Amended) The system of claim 28, wherein the plurality of resources include at least one of a group consisting of: a firewall; data and statistics; storage; ~~compute;~~ a load evaluator; general settings; a routing engine; a virtual machine master controller; a power controller; and a wireless engine.

30. (Currently Amended) The system of claim 19~~, wherein the programmable network device and programmable cloud device further comprise~~ further comprising:

a distributed resource service which is capable of providing at least one service from a group consisting of: exposing application programming interfaces (APIs) to other applications; configuring and managing platform resources; policy enforcement and authorization of the plurality of network device applications and the plurality of cloud applications access to platform resources ~~resource and other application's APIs~~; and policy conflict resolution.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0412

31. (Currently Amended) The system of claim 19~~, wherein the programmable network device and the programmable cloud device further comprise~~ further comprising:

a distributed content service wherein contents generated by the at least one of the plurality of [first] network device applications are shared with [the] at least one of the plurality of ~~second network~~ cloud applications.

32. (Currently Amended) The system of claim 21~~, wherein the flexible network device and flexible cloud platform further comprise~~ further comprising:

a distributed notification service wherein notice of a predetermined event is sent from [the] at least one ~~fxDeviceApp~~ of the plurality of network device applications to [the] at least one of the plurality of cloud applications ~~fxCloudApp~~.

33. (Currently Amended) The system of claim 28, wherein the DRS further includes a load controller~~; and wherein the load controller is~~ adapted to monitor loads on [the] at least one of the plurality of [first] network device applications and at least one of the plurality of ~~second network~~ cloud applications and effect change in accordance with thresholds received from the application management portal.

34. (Currently Amended) The system of claim 19, wherein the programmable network device, at least one of the plurality of [first] network device applications, the programmable cloud device, and the at least one of the plurality of ~~second network~~ cloud applications each have unique identities.

35. (Currently Amended) The system of claim 19, wherein [the] at least one of the plurality of [first] network device applications and at least one of the plurality of ~~second network~~ cloud applications are given an access level to determine which APIs can be accessed.

36. (Currently Amended) The system of claim 19, wherein the ~~at least one~~ programmable network device and the programmable cloud device are adapted to verify the authenticity and integrity of updates to the plurality of network device

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0413

applications and the plurality of cloud applications.

37. (Canceled)

38. (Canceled)

Please add the following new claim:

39. (New) A system comprising:

a programmable network device adapted to host a plurality of network device applications;

a programmable cloud device adapted to host a plurality of cloud applications, wherein the plurality of network device applications and the plurality of cloud applications are in secure communication with each other to form distributed applications;

wherein the plurality of network device applications and plurality of cloud applications form unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) to simultaneously program the plurality of network device applications and plurality of cloud applications independent of network device hardware and cloud device hardware

an application management portal capable of managing life cycles of the plurality of network device applications and plurality of cloud applications; and

an infrastructure application marketplace in communication with the application management portal, said infrastructure marketplace capable of providing tested distributed applications to the application management portal.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0414

REMARKS

Claims 1-18 and 37-38 are canceled. Claim 39 has been added. Claims 19-36 and 39 are pending.

REJECTIONS UNDER 35 USC §112(a)

Claims 19-36 are rejected under 35 USC §112(a). Page 3 of the Office Action states:

> **As per claims 19-36 the limitation "...wherein the programmable network device and programmable cloud device *form unified capabilities enabling a plurality of upper layer application programming interfaces (AP/s) that allow for simultaneous programming of the programmable network device and programmable cloud device independent of network device hardware and cloud device hardware",* does not appearto be described in the specification in such a way as to reasonably convey to one skilledin the relevant art that the inventor or a joint inventor, or for applications subject to pre- AIA 35 U.S.C. 112, the inventor(s), at the time the application was filed, had possession of the claimed invention.**
>
> **Particularly Applicant's specification is completely silent as to the steps or processes for the *"simultaneous programming of the programmable network device and programmable cloud device independent of network device hardware and cloud device hardware".***

The Examiner's attention is pointed to paragraph [0030] of the specification which states (underling added):

> **[0030] Although process (or method) steps may be described or claimed in a particular sequential order, such processes may be configured to work in different orders. Further, some steps may be performed simultaneously despite being described or implied as occurring non-simultaneously (e.g., because one step is described after the other step) unless specifically indicated.**

It is respectfully submitted that the use of "simultaneously" in independent claim 19 (as well as new independent claim 39) is in accordance with paragraph [0030].

REJECTIONS UNDER 35 USC §102(a)(2)

Claims 37-38 are rejected under 35 USC §102(a)(2) as being anticipated by U.S. Patent Publication No. 2013/0054763 to Van der Merwe (hereinafter, "Van der Merwe").

Accordingly, Applicant has canceled claims 37 and 38.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0415

REMARKS

It is respectfully submitted that claims 19-36 and 39 are allowable over the prior art of record. No new matter has been added in the new claims and/or limitations presented in this Amendment.

Respectfully submitted,
ROARK IP
By: /Christopher J. Horgan/

Christopher J. Horgan
Registration No. 40,394
1438 Dahlia Loop
San Jose, CA 95126
Telephone: (408) 687-8306

Date: September 19, 2022

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0416

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 17/142,983	Filing Date 01/06/2021	☐ 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 $50 =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*		x $240 =	
☐ 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	09/19/2022	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 19	Minus	** 20	= 0		x $50 =	0
	Independent (37 CFR 1.16(h))	* 2	Minus	*** 3	= 0		x $240 =	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 $0 =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=		x $0 =	
	☐ 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.	LIE
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".	/SHERRY A DAVIS/
*** 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.	

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.

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.
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3	2943

112449 7590 12/16/2022

Roark IP
1438 Dahlia Loop
San Jose, CA 95126

EXAMINER	
HIGA, BRENDAN Y	

ART UNIT	PAPER NUMBER
2441	

MAIL DATE	DELIVERY MODE
12/16/2022	PAPER

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.

Office Action Summary	Application No. 17/142,983	Applicant(s) Taaghol et al.	
	Examiner BRENDAN HIGA	Art Unit 2441	AIA (FITF) Status Yes

-- 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 09/19/2022.

☐ 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) 19-36 and 39 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) 19-36 and 39 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

This Office action is in response to Applicant's amendment and request for reconsideration filed on 09/19/2022.

Claims 19-36 and 39 are pending.

Claim Objections

Claims 30 and 39 are objected to because there is a typographical error in the claim. For the purpose of this office action the Examiner is interpreting the claim to read:

as per claim 30, "… and authorization of the plurality of network device applications and the plurality of cloud applications ***to*** *access* ***[[to]]*** *platform resources…"; and*

*as per claim 39, "…enabling a plurality of upper layer application programming interfaces (APIs) to simultaneously program the network device applications and plurality of cloud applications independent of network device hardware and cloud device hardware****;***

an application management portal…".

Response to Arguments

Applicant's arguments filed September 19, 2022, with respect to the rejection under 35 U.S.C. 112(a) as being supported by ¶0030 of Applicant's specification, have been fully considered but they are not persuasive.

First, Applicant's specification does not appear to describe specific steps for programming the network device applications and plurality of cloud applications by the

plurality of upper layer application programming interfaces (API). Thus, mentioning that steps of the invention can broadly be performed *"simultaneously"*, as in ¶0030, is insufficient to support the specific limitations of:

"enabling a plurality of upper layer application programming interfaces (APIs) to simultaneously program the network device applications and plurality of cloud applications independent of network device hardware and cloud device hardware".

Secondly, as noted in MPEP §2161.01: *"original claims may lack written description when the claims define the invention in functional language specifying a desired result but the specification does not sufficiently describe how the function is performed or the result is achieved. For software, this can occur when the algorithm or steps/procedure for performing the computer function are not explained at all or are not explained in sufficient detail (simply restating the function recited in the claim is not necessarily sufficient). In other words, the algorithm or steps/procedure taken to perform the function must be described with sufficient detail so that one of ordinary skill in the art would understand how the inventor intended the function to be performed."*

Thus, simply stating that steps can be performed *"simultaneously"* as in Applicant's ¶0030, without describing how the result is achieved - i.e., an algorithm or steps/procedure taken to perform the function - amounts to no more than restating the function recited in the claim, which is insufficient to satisfy the written description requirement of 35 U.S.C. §112(a).

Claim Rejections - 35 USC § 112

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

> (a) IN GENERAL.—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 or joint inventor of carrying out the invention.

The following is a quotation of the first paragraph of pre-AIA 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.

Claims 19-36 and 39 are rejected under 35 U.S.C. 112(a) or 35 U.S.C. 112 (pre-AIA), 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 or a joint inventor, or for applications subject to pre-AIA 35 U.S.C. 112, the inventor(s), at the time the application was filed, had possession of the claimed invention.

As per claims 19-36 and 39 the limitation "…wherein the programmable network device and programmable cloud device *form unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) to simultaneously program the network device applications and plurality of cloud applications independent of network device hardware and cloud device hardware*", does not appear to be described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor or a joint inventor, or for applications subject to pre-AIA 35 U.S.C. 112, the inventor(s), at the time the application was filed, had possession of the claimed invention.

Particularly Applicant's specification is completely silent as to the steps or processes to "*simultaneously program the network device applications and plurality of cloud applications independent of network device hardware and cloud device hardware".*

> The following is a quotation of 35 U.S.C. 112(b):
> (b) CONCLUSION.—The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention.
>
> The following is a quotation of 35 U.S.C. 112 (pre-AIA), second paragraph:
> 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.

Claim 23 is rejected under 35 U.S.C. 112(b) or 35 U.S.C. 112 (pre-AIA), second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the inventor or a joint inventor (or for applications subject to pre-AIA 35 U.S.C. 112, the applicant), regards as the invention.

As per claim 23, the limitation *"the plurality of network device application upgrades"* (i.e., *"…upgrading and commissioning agents for at least one of the plurality of network device application upgrades and related procedures"*) lacks antecedent basis in the claim. For the purpose of this office action the Examiner is interpreting the claim to read:

"…upgrading and commissioning agents for ***upgrading*** *at least one of the plurality of network device applications* ~~***upgrades***~~ *and related procedures"*

As per claim 24, the limitation *"the plurality of cloud application updates"* (i.e., *"…upgrading and commissioning agents for at least one the plurality of cloud*

application updates") lacks antecedent basis in the claim. For the purpose of this office action the Examiner is interpreting the claim to read:

"…upgrading and commissioning agents for ***updating*** *at least one the plurality of cloud application****s*** ~~***updates***~~*"*.

Allowable Subject Matter

Claims 19-36 and 39, though rejected under 35 U.S.C. 112(a), for purposes of 35 U.S.C. 102 and 103, are allowed over the prior art.

Conclusion

THIS ACTION IS MADE FINAL. 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 mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to BRENDAN HIGA whose telephone number is (571)272-5823. The examiner can normally be reached on Monday - Friday 8:30 AM - 5:00 PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at http://www.uspto.gov/interviewpractice.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wing Chan can be reached on (571) 272-7493. 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). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/BRENDAN Y HIGA/
Primary Examiner, Art Unit 2441

Notice of References Cited	Application/Control No. 17/142,983	Applicant(s)/Patent Under Reexamination Taaghol et al.	
	Examiner BRENDAN HIGA	Art Unit 2441	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20170257432-A1	09-2017	FU; TIANYING	H04L67/1014	1/1
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC 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.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0426

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	Art Unit
	BRENDAN HIGA	2441

CPC - Searched*		
Symbol	Date	Examiner
H04L41/08,0895,12-122,22; 45/00,036,56-563,586,64,645,655,76; 67/34	12/14/2022	BH

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
see SEARCH history	12/14/2022	BH
IP.com (NPL+Patents)	12/14/2022	BH
Google Scholar (see search history and results)	12/14/2022	BH
Reviewed prosecution history of US application 14/295331, particularly final rejection mailed on 12/28/2016	09/02/2022	BH
PALM inventor search	12/14/2022	BH

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
H04L	41/08,0895,12-122,22; 45/00,036,56-563,586,64,645,655,76; 67/34	12/14/2022	BH

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0427

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	Art Unit
	BRENDAN HIGA	2441

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0428

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	**Art Unit**
	BRENDAN HIGA	2441

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

CLAIMS

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

CLAIM		DATE								
Final	Original	01/01/2022	04/07/2022	09/04/2022	12/14/2022					
	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			✓	✓					
	37			✓	-					
	38			✓	-					
	39			✓	✓					

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0429

IQSearchText-202212142148

Main Concept Document
==========
US10893095
Title: Distributed software defined networking

(No Concept Modifiers)
==========

Filters (1)
==========
Publication Date: To 2013-12-31

Options
==========
De-dup: Extended Family Numbers (PAT only)
Relevance Cut-off: None
Sort: Relevance

Content: Patents + NPL (192)
==========
US Patents
US Designs
US Applications
EPO Patents
EPO Applications
China Patents
China Applications
Japan Patents
Japan Applications
Korea Patents
Korea Applications
WIPO Applications
Argentina Patents
Argentina Applications
Brazil Patents
Brazil Applications
Canada Patents
Canada Applications
Chile Patents
Chile Applications
Colombia Applications
Costa Rica Applications
Cuba Patents

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0430

Cuba Applications
Dominican Republic Applications
Ecuador Patents
Ecuador Applications
El Salvador Applications
Guatemala Applications
Honduras Applications
Mexico Patents
Mexico Applications
Nicaragua Patents
Panama Applications
Peru Applications
Trinidad & Tobago Patents
Uruguay Applications
Austria Patents
Austria Applications
Belarus Patents
Belgium Patents
Belgium Applications
Bosnia & Herzegovina Patents
Bosnia & Herzegovina Applications
Bulgaria Patents
Bulgaria Applications
Croatia Patents
Croatia Applications
Czech Republic Patents
Czech Republic Applications
Czechoslovakia Patents
Czechoslovakia Applications
Denmark Patents
Denmark Applications
Estonia Patents
Estonia Applications
EUIPO Patents
Finland Patents
Finland Applications
France Patents
France Applications
Germany Patents
Germany Applications
Great Britain Patents
Great Britain Applications
Greece Patents
Greece Applications
Hungary Patents
Hungary Applications
Iceland Patents
Iceland Applications

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0431

Ireland Patents
Ireland Applications
Italy Patents
Italy Applications
Latvia Patents
Latvia Applications
Lithuania Patents
Lithuania Applications
Luxembourg Patents
Luxembourg Applications
Malta Patents
Monaco Patents
Montenegro Patents
Montenegro Applications
Netherlands Patents
Netherlands Applications
Norway Patents
Norway Applications
Poland Patents
Poland Applications
Portugal Patents
Portugal Applications
Republic of Moldova Patents
Republic of Moldova Applications
Romania Patents
Romania Applications
San Marino Patents
San Marino Applications
Serbia Patents
Serbia Applications
Slovakia Patents
Slovakia Applications
Slovenia Patents
Spain Patents
Spain Applications
Sweden Patents
Sweden Applications
Switzerland Patents
Switzerland Applications
Ukraine Patents
Yugoslavia/Serbia and Montenegro Patents
Yugoslavia/Serbia and Montenegro Applications
Armenia Patents
Australia Patents
Australia Applications
Cyprus Patents
Gulf Cooperation Council Patents
Hong Kong Patents

India Patents
India Applications
Indonesia Patents
Indonesia Applications
Israel Patents
Israel Applications
Jordan Patents
Jordan Applications
Kyrgyzstan Patents
Macao Applications
Malaysia Patents
Mongolia Patents
New Zealand Patents
Philippines Patents
Philippines Applications
Saudi Arabia Patents
Saudi Arabia Applications
Singapore Patents
Singapore Applications
Taiwan Patents
Taiwan Applications
Tajikistan Patents
Tajikistan Applications
Thailand Patents
Thailand Applications
Uzbekistan Patents
Vietnam Patents
Algeria Patents
ARIPO Patents
ARIPO Applications
Egypt Patents
Kenya Patents
Malawi Patents
Morocco Patents
Morocco Applications
OAPI Patents
South Africa Patents
Tunisia Applications
Zambia Patents
Zimbabwe Patents
EAPO Patents
EAPO Applications
Georgia Patents
Georgia Applications
Kazakhstan Patents
Kazakhstan Applications
Russia Patents
Russia Applications

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0433

Turkey Patents
Turkey Applications
IEEE Xplore Publications: IEEE Periodicals
IEEE Xplore Publications: IEEE Conferences
IEEE Xplore Publications: IEEE Standards
IEEE Xplore Publications: IEEE Early Access
IEEE Xplore Publications: SMPTE Periodicals
IEEE Xplore Publications: SMPTE Conferences
IEEE Xplore Publications: SMPTE Standards
IEEE Xplore Publications: MIT Press eBooks
IEEE Xplore Publications: Wiley-IEEE eBooks
IEEE Xplore Publications: IBM Periodicals
IEEE Xplore Publications: URSI Periodicals
IEEE Xplore Publications: VDE Conferences
IEEE Xplore Publications: Periodicals from China
IP.com Prior Art Database: The IP.com Journal
IP.com Prior Art Database: Internet Society RFC
IP.com Prior Art Database: IBM TDB Archive
IP.com Prior Art Database: Legacy Journals
IP.com Prior Art Database: Software Patent Institute
OnePetro.org: Periodicals at OnePetro.org
OnePetro.org: Conferences at OnePetro.org
Other Literature: IBM Redbooks
Other Literature: PubMed Central
Other Literature: arXiv.org

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0434

Google Scholar | simultaneous programming of distributed applications across a cloud infrastructure | SIGN IN

Articles | About 20,800 results (0.18 sec) | My profile | My library

Any time
Since 2022
Since 2021
Since 2018
Custom range...
— 2013

Sort by relevance
Sort by date

Any type
Review articles

include patents
include citations

Create alert

[BOOK] **Distributed** and **cloud computing**: from parallel processing to the internet of things — [PDF] pace.edu
K Hwang, J Dongarra, GC Fox - 2013 - books.google.com
... cover **programming**, and the use of **distributed** or **cloud** systems in innovative Internet **applications**... A typical P2P network may involve millions of client machines working ...
Save Cite Cited by 606 Related articles All 8 versions

Cloudanalyst: A cloudsim-based visual modeller for analysing **cloud computing** environments and **applications** — [PDF] ieee.org Full Text @ USPTO-STIC
B Wickremasinghe, RN Calheiros... - ... and applications, 2010 - ieeexplore.ieee.org
... to **distribute applications** among **Cloud infrastructures** and value added services such as optimization of **applications** ... Quantifying impact of number of **simultaneous** users, geographic ...
Save Cite Cited by 769 Related articles All 12 versions

[BOOK] Mastering **cloud computing**: foundations and **applications programming**
R Buyya, C Vecchiola, ST Selvi - 2013 - books.google.com
... **applications** for PCs to **cloud** datacenters that enable millions of users to use software **simultaneously**... to the concepts of parallel and **distributed computing**, Chapter 2 can be skipped. ...
Save Cite Cited by 491 Related articles All 5 versions

Load balancing in **cloud computing** using stochastic hill climbing-a soft **computing** approach — [PDF] sciencedirect.com Full Text @ USPTO-STIC
B Mondal, K Dasgupta, P Dutta - Procedia Technology, 2012 - Elsevier
... in **cloud computing** provides an organization with the ability to **distribute application** requests **across** ... Quantifying impact of number of **simultaneous** users, geographic location of relevant ...
Save Cite Cited by 277 Related articles All 6 versions

[PDF] Availability and load balancing in **cloud computing** — [PDF] academia.edu
Z Chaczko, V Mahadevan, S Aslanzadeh... - ... on Computer and ..., 2011 - academia.edu
... , **across** multiple availability zones, reduces the chance of outages that could **simultaneously** affect the services in **cloud** ... to use **distributed applications through** different computers on ...
Save Cite Cited by 271 Related articles All 5 versions

Multimedia **cloud computing** — [PDF] ieee.org Full Text @ USPTO-STIC
W Zhu, C Luo, J Wang, S Li - IEEE Signal Processing ... , 2011 - ieeexplore.ieee.org
... users' **simultaneous** accesses at the Internet scale. In today's **cloud computing**, the **cloud** uses a ... P2P multimedia computing refers to a **distributed application** architecture that partitions ...
Save Cite Cited by 564 Related articles All 3 versions

[PDF] Above the **clouds**: A berkeley view of **cloud computing** — [PDF] ust.hk
M Armbrust, A Fox, R Griffith, AD Joseph, RH Katz... - 2009 - cse.ust.hk
... Hours purchased via **Cloud Computing** can be **distributed** ... **computing programmers** to rewrite their **programs** using more ... be for the **Cloud Computing** provider to offer **simultaneous** gang ...
Save Cite Cited by 8639 Related articles All 122 versions

Modeling and simulation of scalable **Cloud computing** environments and the CloudSim toolkit: Challenges and opportunities — [PDF] ieee.org Full Text @ USPTO-STIC

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0435

R Buyya, R Ranjan... - ... performance computing & ..., 2009 - ieeexplore.ieee.org
... SimGrid is a generic framework for simulation of **distributed applications** on Grid platforms. ... As discussed earlier, a host can **simultaneously** instantiate multiple VMs and allocate cores ...
Save Cite Cited by 1525 Related articles All 16 versions

A survey of mobile **cloud computing**: architecture, **applications**, and approaches [PDF] wiley.com Full View
HT Dinh, C Lee, D Niyato... - ... and mobile computing, 2013 - Wiley Online Library
... sharing files and folders **across** multiple devices **simultaneously**. ... **distributed** partitioning system called Coign, which automatically transforms a **program** into the **distributed applications** ...
Save Cite Cited by 3012 Related articles All 11 versions

[PDF] Above the **clouds**: A berkeley view of **cloud computing** [PDF] academia.edu
A Fox, R Griffith, A Joseph, R Katz... - Dept. Electrical Eng ..., 2009 - academia.edu
... Hours purchased via **Cloud Computing** can be **distributed** ... **computing programmers** to rewrite their **programs** using more ... be for the **Cloud Computing** provider to offer **simultaneous** gang ...
Save Cite Cited by 892 Related articles All 11 versions

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0436

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L1	0	17/142983.app.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/12/14 02:45 PM
L2	282077	(H04L41/08,0895,12-122,22 H04L45/00,036,56-563,586,64,645,655,76 H04L67/34).cpc.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/12/14 04:28 PM
L3	329726	(cloud docker kubernetes SDN NFV Open$1flow open$1stack VM virtual$4) adj15 (install$6 deploy$5 orchestrat$6)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/12/14 04:29 PM
L4	486189	(Program$6 API) adj10 (cloud docker kubernetes SDN NFV Open$1flow	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU,	OR	ON	ON	2022/12/14 04:31 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0437

		open$1stack VM virtual$4)	CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L5	17715	(simultaneous$2 concurrent$3 synchronous$3) SAME 4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/12/14 04:33 PM
L6	291	2 AND 3 AND 5	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/12/14 04:33 PM
L7	40	6 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN,	OR	ON	ON	2022/12/14 04:33 PM

			WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L8	185398	(cloud docker kubernetes SDN NFV Open$1flow open$1stack VM virtual$4) adj15 (deploy$5 orchestrat$6)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/12/14 04:35 PM
L9	6273	2 AND 4 AND 8	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/12/14 04:35 PM
L10	3037	2 AND 4 SAME2 8	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/12/14 04:35 PM
L11	39257	(Program$6 API) adj15 (distribut$4 ADJ5 (service application app))	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB,	OR	ON	ON	2022/12/14 04:37 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0439

			HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L12	121	11 AND 10	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/12/14 04:37 PM
L13	34	12 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/12/14 04:44 PM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N1	692	(simultaneous$2 concurrent$3 synchronous$3).CLM. SAME (Program$6 API) adj10 (cloud docker kubernetes SDN NFV Open$1flow open$1stack VM virtual$4).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2022/12/14 04:30 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0440

N2	72329	(H04L41/08,0895,12-122,22 H04L45/00,036,56-563,586,64,645,655,76 H04L67/34).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2022/12/14 04:30 PM
N3	21	N1 AND N2	(US-PGPUB; USPAT)	OR	ON	ON	2022/12/14 04:30 PM
N4	7	N3 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT)	OR	ON	ON	2022/12/14 04:30 PM
N5	183382	(Program$6 API) adj10 (cloud docker kubernetes SDN NFV Open$1flow open$1stack VM virtual$4)	(US-PGPUB; USPAT)	OR	ON	ON	2022/12/14 04:31 PM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0441

Web Search History

date, time	web site	search string
12/14/2022 4:17:46 PM	Google Scholar	programmable cloud infrastructure for wireless or mobile or cellular network
12/14/2022 4:17:53 PM	Google Scholar	programmable cloud infrastructure for wireless or mobile or cellular network [before:2013]
12/14/2022 4:18:09 PM	Google Scholar	[before:2013] programmable cloud infrastructure for wireless or mobile or cellular network and "API"
12/14/2022 4:51:53 PM	Google Scholar	[before:2013] simultaneous programmable cloud infrastructure for wireless or mobile or cellular network and "API"
12/14/2022 4:52:59 PM	Google Scholar	[before:2013] APIs for programming distributed applications in a cloud infrastructure
12/14/2022 4:53:41 PM	Google Scholar	[before:2013] simultaneous programming of distributed applications across a cloud infrastructure

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0442

UNITED STATES PATENT AND TRADEMARK OFFICE

USPTO Automated Interview Request (AIR)

Mar 14 2023

This paper requesting to schedule and/or conduct an interview is appropriate because:

I have previously submitted an authorization for internet communications in a patent application. See form SB/0439.

Name(s):
Christopher Horgan

S-signature:
/Christopher Horgan/

Registration Number:
40394

U.S. Application Number:
17142983

Confirmation Number:
2943

E-mail Address:
chris.horgan@roarkip.com

Phone Number:
+1 4086878306

Proposed Time of Interview:
3-21-2023 10:30 AM ET

Alternative Proposed Time(s) of Interview:
3-23-2023 10:30 AM ET

Alternative Proposed Time(s) of Interview:
3-24-2023 10:30 AM ET

Prefered Interview Type:
Telephonic

I am the applicant or applicant's representative for this application.

Topic for Discussion:
35 USC 112 rejections

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.
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3	2943

112449 7590 03/20/2023

Roark IP
1438 Dahlia Loop
San Jose, CA 95126

EXAMINER	
HIGA, BRENDAN Y	

ART UNIT	PAPER NUMBER
2441	

MAIL DATE	DELIVERY MODE
03/20/2023	PAPER

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.

Applicant-Initiated Interview Summary	**Application No.** 17/142,983	**Applicant(s)** Taaghol et al.		
	Examiner BRENDAN HIGA	**Art Unit** 2441	**AIA (First Inventor to File) Status** Yes	**Page** 1 of 1

All Participants (applicant, applicants representative, PTO personnel)	**Title**	**Type**
BRENDAN HIGA	Primary Examiner	Telephonic
Christopher Hogan (Reg. No. 40,394)	Attorney	

Date of Interview: 16 March 2023

Issues Discussed:

35 U.S.C. 112

Discussed the filing of a continuation and whether removing "simultaneously" from the limitation "wherein the plurality of network device applications and plurality of cloud applications form unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) to simultaneously program the plurality of network device applications and plurality of cloud applications", would be sufficient to overcome the 112(a) rejection. The Examiner indicated that further consideration would be required in determining whether the specification recites a sufficient algorithm (e.g., the necessary steps and/or flowcharts, see MPEP 2161.01) for meeting the written description requirement for the computer-implemented "enabling" function to be performed.

/BRENDAN Y HIGA/ Primary Examiner, Art Unit 2441	

Applicant is reminded that a complete written statement as to the substance of the interview must be made of record in the application file. It is the applicants responsibility to provide the written statement, unless the interview was initiated by the Examiner and the Examiner has indicated that a written summary will be provided. See MPEP 713.04
Please further see:
MPEP 713.04
Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews, paragraph (b)
37 CFR § 1.2 Business to be transacted in writing

Applicant recordation instructions: The formal written reply to the last Office action must include the substance of the interview. (See MPEP section 713.04). If a reply to the last Office action has already been filed, applicant is given a non-extendable period of the longer of one month or thirty days from this interview date, or the mailing date of this interview summary form, whichever is later, to file a statement of the substance of the interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0445

Electronic Patent Application Fee Transmittal

Application Number:	17142983
Filing Date:	06-Jan-2021
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Filer:	Christopher James Horgan
Attorney Docket Number:	TAAG001CONT3

Filed as Small Entity

Filing Fees for Utility under 35 USC 111(a)

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:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension - 1 month with $0 paid	2251	1	88	88
Miscellaneous:				
		Total in USD ($)		**88**

Electronic Acknowledgement Receipt

EFS ID:	47847083
Application Number:	17142983
International Application Number:	
Confirmation Number:	2943
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Customer Number:	112449
Filer:	Christopher James Horgan
Filer Authorized By:	
Attorney Docket Number:	TAAG001CONT3
Receipt Date:	14-APR-2023
Filing Date:	06-JAN-2021
Time Stamp:	17:04:18
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$88
RAM confirmation Number	E20234DH06386861
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Fee Worksheet (SB06)	fee-info.pdf	38108 10c02639a5bb4fca24514e74c8ed3390822fc202	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	38108		

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.

PTO/SB/30 (11-17)
Approved for use through 05/31/2024. 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

Address to:
Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Application Number	17/142,983
Filing Date	1/6/2021
First Named Inventor	Taaghol
Art Unit	2441
Examiner Name	Higa
Attorney Docket Number	TAAG001CONT3

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. See Instruction Sheet for RCEs (not to be submitted to the USPTO) on page 2.

1. 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).

   a. ☐ 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.

      i. ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____

      ii. ☐ Other _____

   b. ☑ Enclosed

      i. ☑ Amendment/Reply

      ii. ☐ Affidavit(s)/ Declaration(s)

      iii. ☐ Information Disclosure Statement (IDS)

      iv. ☐ Other _____

2. Miscellaneous

   a. ☐ 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)

   b. ☐ Other _____

3. Fees The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.

   a. ☐ The Director is hereby authorized to charge the following fees, any underpayment of fees, or credit any overpayments, to Deposit Account No. _____.

      i. ☐ RCE fee required under 37 CFR 1.17(e)

      ii. ☐ Extension of time fee (37 CFR 1.136 and 1.17)

      iii. ☐ Other _____

   b. ☐ Check in the amount of $ _____ enclosed

   c. ☐ Payment by credit card (Form PTO-2038 enclosed)

   d. ☐ Payment by EFS-Web

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

Signature	/Christopher J. Horgan/	Date	April 9, 2023
Name (Print/Type)	Christopher J. Horgan	Registration No.	40,394

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being EFS-Web transmitted to the United States Patent and Trademark Office (USPTO), deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop RCE, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450 or facsimile transmitted to the USPTO on the date shown below.

Signature			
Name (Print/Type)		Date	

A Federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with an information collection subject to the requirements of the Paperwork Reduction Act of 1995, unless the information collection has a currently valid OMB Control Number. The OMB Control Number for this information collection is 0651-0031. Public burden for this form is estimated to average 12 minutes per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the information collection. Send comments regarding this burden estimate or any other aspect of this information collection, including suggestions for reducing this burden to the Chief Administrative Officer, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 or email InformationCollection@uspto.gov. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **If filing this completed form by mail, 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.

Electronic Patent Application Fee Transmittal

Application Number:	17142983
Filing Date:	06-Jan-2021
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Filer:	Christopher James Horgan
Attorney Docket Number:	TAAG001CONT3

Filed as Small Entity

Filing Fees for Utility under 35 USC 111(a)

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:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
RCE- 2ND AND SUBSEQUENT REQUEST	2820	1	800	800
		Total in USD ($)		**800**

Electronic Acknowledgement Receipt

EFS ID:	47855141
Application Number:	17142983
International Application Number:	
Confirmation Number:	2943
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Customer Number:	112449
Filer:	Christopher James Horgan
Filer Authorized By:	
Attorney Docket Number:	TAAG001CONT3
Receipt Date:	17-APR-2023
Filing Date:	06-JAN-2021
Time Stamp:	17:21:52
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$800
RAM confirmation Number	E20234GH24068212
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Continued Examination (RCE)	TAAG001CONT3_RCE.pdf	265768 5fd6e0eceec57770f6bd3cd897dcebd46b517ca9	no	1
Warnings:					
This is not a USPTO supplied RCE SB30 form.					
Information:					
2	Amendment Submitted/Entered with Filing of Continued Prosecution Application (CPA)/Request for Continued Examination(RCE)	TAAG001CONT3_RESPONSE_APR2023.pdf	66637 75b110945cbd66334153cea12e0c34dc484b514f	no	10
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	38177 570fa0d3be697be1b417e68eea38f3916a80464b	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	370582		

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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Taaghol
Serial No.: 17/142,983
Filed: 1/6/2021
Attorney Docket No.: TAAG001CONT3

Examiner: Higa
Art Unit: 2441

For: DISTRIBUTED SOFTWARE DEFINED NETWORKING

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

Dear Sir:

RESPONSE

Applicants offer the following election in response to the Office Action (hereinafter "Office Action") mailed on December 16, 2022.

Amendments to the Claims begin on page 2 of this paper.
Applicant's Statement of Substance of Interview begins on page 8 of this paper.
Remarks begin on page 9 of this paper.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0455

In The Claims:

1. (Canceled)
2. (Canceled)
3. (Canceled)
4. (Canceled)
5. (Canceled)
6. (Canceled)
7. (Canceled)
8. (Canceled)
9. (Canceled)
10. (Canceled)
11. (Canceled)
12. (Canceled)
13. (Canceled)
14. (Canceled)
15. (Canceled)
16. (Canceled)
17. (Canceled)
18. (Canceled)

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0456

19. (Currently Amended) A system comprising:

a programmable network device adapted to host a plurality of network device applications;

a programmable cloud device adapted to host a plurality of cloud applications, wherein the plurality of network device applications and the plurality of cloud applications are in secure communication with each other to form distributed applications; and

wherein the plurality of network device applications and plurality of cloud applications device form unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) to ~~simultaneously~~ program the plurality of network device applications and plurality of cloud applications independent of network device hardware and cloud device hardware.

20. (Previously Presented) The system of claim 19, wherein the programmable network device and the programmable cloud device each include an access network interface unit configured to send and receive communications and a processor with a memory associated with the access network interface unit.

21. (Previously Presented) The system of claim 19 further comprising:

an application management portal capable of managing life cycles of the distributed applications; and

an infrastructure application marketplace in communication with the application management portal, said infrastructure marketplace capable of providing tested and certified distributed applications to the application management portal.

22. (Currently Amended) The system of claim 21, wherein the application management portal is capable of at least one of the group consisting of: receiving new applications from the infrastructure application marketplace; testing said distributed applications prior to deployment; ~~provision~~ provisioning said distributed applications; and ~~deprovision~~ deprovisioning said distributed applications.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0457

23. (Currently Amended) The system of claim 19, wherein at least one of the plurality of network device applications is responsible for a plurality of functional capabilities of the programmable network device and is capable of at least one of the group consisting of: updating firmware of the programmable network device; updating default communication protocols of the programmable network device; initializing programmable network device settings; boot loading the programmable network device; providing security digital certificates; providing operating system (OS) and kernel upgrades; and upgrading and commissioning agents for upgrading at least one of the plurality of network device applications ~~application upgrades~~ and related procedures.

24. (Currently Amended) The system of claim 19, wherein at least one of the plurality of cloud applications is responsible for a plurality of functional capabilities of the programmable cloud device and is capable of at least one of the group consisting of: upgrading and commissioning agents for updating at least one of the plurality of cloud applications ~~application updates~~; hypervisor upgrades; virtual machine and isolated execution environment lifecycle management; security digital certificate management; operating system (OS) and kernel upgrades; and default communication protocols.

25. (Previously Presented) The system of claim 19, wherein the programmable network device is an LTE eNB and the programmable cloud device hosts LTE core network functions in separate virtual machines from the group consisting of: MME, S-GW, and P-GW.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0458

26. (Previously Presented) The system of claim 19, further comprising:

a virtual fabric which provides a secure communication layer for said at least one of the plurality of network device applications and said at least one of the plurality of cloud applications.

27. (Previously Presented) The system of claim 19, further comprising:

a second programmable network device which includes at least one of the plurality of network device applications which is the same as at least one of the plurality of network device applications in the programmable network device and where each of the network device applications can communicate directly.

28. (Previously Presented) The system of claim 21, further comprising:

a distributed resource service (DRS) which controls access to a plurality of resources upon receiving instructions from the application management portal.

29. (Previously Presented) The system of claim 28, wherein the plurality of resources include at least one of a group consisting of: a firewall; data and statistics; storage; a load evaluator; general settings; a routing engine; a virtual machine master controller; a power controller; and a wireless engine.

30. (Currently Amended) The system of claim 19, further comprising:

a distributed resource service which is capable of providing at least one service from a group consisting of: exposing application programming interfaces (APIs) to other applications; configuring and managing platform resources; policy enforcement and authorization of the plurality of network device applications and the plurality of cloud applications to access [to] platform resources; and policy conflict resolution.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0459

31. (Previously Presented) The system of claim 19 further comprising:

a distributed content service wherein contents generated by the at least one of the plurality of network device applications are shared with at least one of the plurality of cloud applications.

32. (Previously Presented) The system of claim 21 further comprising:

a distributed notification service wherein notice of a predetermined event is sent from at least one of the plurality of network device applications to at least one of the plurality of cloud applications.

33. (Previously Presented) The system of claim 28, wherein the DRS further includes a load controller adapted to monitor loads on at least one of the plurality of network device applications and at least one of the plurality of cloud applications and effect change in accordance with thresholds received from the application management portal.

34. (Previously Presented) The system of claim 19, wherein the programmable network device, at least one of the plurality of network device applications, the programmable cloud device, and the at least one of the plurality of cloud applications each have unique identities.

35. (Previously Presented) The system of claim 19, wherein at least one of the plurality of network device applications and at least one of the plurality of cloud applications are given an access level to determine which APIs can be accessed.

36. (Previously Presented) The system of claim 19, wherein the at least one programmable network device and the programmable cloud device are adapted to verify the authenticity and integrity of updates to the plurality of network device applications and the plurality of cloud applications.

37. (Canceled)

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0460

38. (Canceled)

39. (Currently Amended) A system comprising:

a programmable network device adapted to host a plurality of network device applications;

a programmable cloud device adapted to host a plurality of cloud applications, wherein the plurality of network device applications and the plurality of cloud applications are in secure communication with each other to form distributed applications;

wherein the plurality of network device applications and plurality of cloud applications form unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) to ~~simultaneously~~ program the plurality of network device applications and plurality of cloud applications independent of network device hardware and cloud device hardware;

an application management portal capable of managing life cycles of the plurality of network device applications and plurality of cloud applications; and

an infrastructure application marketplace in communication with the application management portal, said infrastructure marketplace capable of providing tested distributed applications to the application management portal.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0461

Applicant's Statement of Substance of Interview

Applicant thanks the Examiner for the courtesy of the telephone interview on March 16, 2023. No specific agreement was reached on overcoming the outstanding rejection.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0462

REMARKS

Claims 1-18 and 37-38 are canceled. Claims 19-36 and 39 are pending.

CLAIM OBJECTIONS

Claims 30 and 39 have been objected to. Accordingly, Applicant has amended claims 30 and 39.

REJECTIONS UNDER 35 USC §112(a)

Claims 19-36 and 39 are rejected under 35 USC §112(a). Accordingly, Applicant has amended the independent claims 19 and 39.

REJECTIONS UNDER 35 USC §112(b)

Claims 23 and 24 are rejected under 35 USC §112(b). Accordingly, Applicant has amended claims 23 and 24.

ALLOWABLE SUBJECT MATTER

Applicant appreciates the indication of allowable subject matter in claims 19-36 and 39.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0463

CONCLUSION

It is respectfully submitted that claims 19-36 and 39 are allowable over the prior art of record. No new matter has been added in the new claims and/or limitations presented in this Amendment.

Respectfully submitted,
ROARK IP
By: /Christopher J. Horgan/

Christopher J. Horgan
Registration No. 40,394
1438 Dahlia Loop
San Jose, CA 95126
Telephone: (408) 687-8306

Date: April 9, 2023

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0464

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 17/142,983	Filing Date 01/06/2021	☐ 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 $50 =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*		x $240 =	
☐ 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	04/17/2023	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 19	Minus	** 20	= 0		x $40 =	0
	Independent (37 CFR 1.16(h))	* 2	Minus	*** 3	= 0		x $192 =	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 $0 =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=		x $0 =	
	☐ 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.	LIE
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".	/TAMMY D MCBETH/
*** 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.	

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.

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

112449 7590 05/09/2023

Roark IP
1438 Dahlia Loop
San Jose, CA 95126

EXAMINER
HIGA, BRENDAN Y

ART UNIT	PAPER NUMBER
2441	

DATE MAILED: 05/09/2023

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3	2943

TITLE OF INVENTION: Distributed Software Defined Networking

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0.00	$0.00	$480	08/09/2023

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 40% the amount of undiscounted fees, and micro entity fees are 20% the amount of undiscounted 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. 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: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0466

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (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 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. **Because electronic patent issuance may occur shortly after issue fee payment, any desired continuing application should preferably be filed prior to payment of this issue fee in order not to jeopardize copendency.**

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

112449 7590 05/09/2023
Roark IP
1438 Dahlia Loop
San Jose, CA 95126

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 transmitted to the USPTO via EFS-Web or by facsimile to (571) 273-2885, on the date below.

(Typed or printed name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3	2943

TITLE OF INVENTION: Distributed Software Defined Networking

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0.00	$0.00	$480	08/09/2023

EXAMINER	ART UNIT	CLASS-SUBCLASS
HIGA, BRENDAN Y	2441	709-201000

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/AIA/122 or PTO/SB/122) attached.

❑ "Fee Address" indication (or "Fee Address" Indication form PTO/AIA/47 or 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 must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). 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. Fees submitted: ❑Issue Fee ❑Publication Fee (if required)

4b. Method of Payment: *(Please first reapply any previously paid fee shown above)*

❑ Electronic Payment via Patent Center or EFS-Web ❑ Enclosed check ❑ Non-electronic payment by credit card (Attach form PTO-2038)

❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. _____

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. _____

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0467

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.
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3	2943

112449 7590 05/09/2023

Roark IP
1438 Dahlia Loop
San Jose, CA 95126

EXAMINER	
HIGA, BRENDAN Y	
ART UNIT	PAPER NUMBER
2441	

DATE MAILED: 05/09/2023

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)

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0468

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 30 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.

Notice of Allowability	Application No. 17/142,983	Applicant(s) Taaghol et al.	
	Examiner BRENDAN HIGA	Art Unit 2441	AIA (FITF) Status Yes

-- 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 Applicant's 04/17/2023 Request for Continued Examination (RCE).
   ☐ 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 19-36 and 39 . 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 _____.

/BRENDAN Y HIGA/
Primary Examiner, Art Unit 2441

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0470

Notice of References Cited	Application/Control No. 17/142,983	Applicant(s)/Patent Under Reexamination Taaghol et al.	
	Examiner BRENDAN HIGA	Art Unit 2441	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20140351428-A1	11-2014	Madani; Habib	H04L47/76	709/224
*	B	US-20140082156-A1	03-2014	Jagtap; Sameer S.	G06F9/5027	709/220
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC 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
	17/142,983	Taaghol et al.
	Examiner	Art Unit
	BRENDAN HIGA	2441

CPC - Searched*		
Symbol	Date	Examiner
H04L41/08,0895; 45/563; 67/10, 34	05/04/2023	BH

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
see SEARCH history	05/04/2023	BH
IP.com (NPL+Patents)	05/04/2023	BH
Google Scholar (see search history and results)	05/04/2023	BH
Reviewed prosecution history of US application 14/295331, particularly final rejection mailed on 12/28/2016	09/02/2022	BH
PALM inventor search	12/14/2022	BH

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
H04L	41/08,0895; 45/563; 67/10, 34	05/04/2023	BH

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0472

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	**Art Unit**
	BRENDAN HIGA	2441

CPC						
Symbol					**Type**	**Version**
H04L	/	67	/	10	F	2013-01-01
G06F	/	21	/	57	I	2013-01-01
G06F	/	21	/	44	I	2013-01-01
H04L	/	67	/	34	I	2013-01-01
H04L	/	41	/	0803	I	2013-01-01
H04L	/	63	/	0227	I	2013-01-01
G06F	/	21	/	53	I	2013-01-01
H04L	/	9	/	0822	I	2013-01-01
H04L	/	41	/	0893	I	2013-01-01
H04L	/	45	/	563	I	2013-01-01
H04L	/	45	/	64	I	2013-01-01
H04L	/	67	/	1076	I	2013-01-01
H04L	/	41	/	0895	I	2022-05-01
H04W	/	4	/	00	A	2013-01-01
H04L	/	63	/	0218	A	2013-01-01
G06F	/	8	/	60	A	2013-01-01
H04L	/	41	/	082	A	2013-01-01
H04L	/	41	/	5019	A	2013-01-01
G06F	/	8	/	70	A	2013-01-01
G06F	/	2009	/	45587	A	2013-01-01

CPC Combination Sets							
Symbol					**Type**	**Set**	**Ranking**	**Version**
	/		/					

NONE		Total Claims Allowed:	
(Assistant Examiner)	(Date)	19	
/BRENDAN Y HIGA/ Primary Examiner, Art Unit 2441	04 May 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	3

U.S. Patent and Trademark Office — Part of Paper No.: 20230426-A

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0473

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	**Art Unit**
	BRENDAN HIGA	2441

INTERNATIONAL CLASSIFICATION				
CLAIMED				
H04L67/10	/	67	/	10
H04L67/00	/	67	/	00
H04L41/0803	/	41	/	0803
H04L9/40	/	9	/	40
H04L9/08	/	9	/	08
H04L41/0893	/	41	/	0893
H04L45/00	/	45	/	00
H04L45/64	/	45	/	64
H04L67/1074	/	67	/	1074
H04L41/0895	/	41	/	0895
H04W4/00	/	4	/	00
H04L41/082	/	41	/	082

NON-CLAIMED				
G06F21/57	/	21	/	57
G06F21/44	/	21	/	44
G06F21/53	/	21	/	53
G06F8/60	/	8	/	60
G06F8/70	/	8	/	70

US ORIGINAL CLASSIFICATION	
CLASS	**SUBCLASS**

CROSS REFERENCES(S)						
CLASS	**SUBCLASS (ONE SUBCLASS PER BLOCK)**					

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	19	
/BRENDAN Y HIGA/ Primary Examiner, Art Unit 2441	04 May 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	3

U.S. Patent and Trademark Office — Part of Paper No.: 20230426-A

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0474

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	**Art Unit**
	BRENDAN HIGA	2441

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

CLAIMS

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	19	5	28	19	39										
2	20	6	29												
3	21	14	30												
4	22	15	31												
9	23	8	32												
10	24	7	33												
11	25	16	34												
12	26	17	35												
13	27	18	36												

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	19	
/BRENDAN Y HIGA/ Primary Examiner, Art Unit 2441	04 May 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	3

U.S. Patent and Trademark Office — Part of Paper No.: 20230426-A

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0475

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/142,983	Taaghol et al.
	Examiner	**Art Unit**
	BRENDAN HIGA	2441

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

CLAIMS

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

CLAIM		DATE								
Final	Original	01/01/2022	04/07/2022	09/04/2022	12/14/2022	05/04/2023				
	1	✓	✓	-	-	-				
	2	✓	✓	-	-	-				
	3	✓	✓	-	-	-				
	4	✓	✓	-	-	-				
	5	✓	✓	-	-	-				
	6	✓	✓	-	-	-				
	7	✓	✓	-	-	-				
	8	✓	✓	-	-	-				
	9	✓	✓	-	-	-				
	10	✓	✓	-	-	-				
	11	✓	✓	-	-	-				
	12	✓	✓	-	-	-				
	13	✓	✓	-	-	-				
	14	✓	✓	-	-	-				
	15	✓	✓	-	-	-				
	16	✓	✓	-	-	-				
	17	✓	✓	-	-	-				
	18	✓	✓	-	-	-				
1	19			✓	✓	=				
2	20			✓	✓	=				
3	21			✓	✓	=				
4	22			✓	✓	=				
9	23			✓	✓	=				
10	24			✓	✓	=				
11	25			✓	✓	=				
12	26			✓	✓	=				
13	27			✓	✓	=				
5	28			✓	✓	=				
6	29			✓	✓	=				
14	30			✓	✓	=				
15	31			✓	✓	=				
8	32			✓	✓	=				
7	33			✓	✓	=				
16	34			✓	✓	=				
17	35			✓	✓	=				
18	36			✓	✓	=				
	37			✓	-	-				
	38			✓	-	-				
19	39			✓	✓	=				

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0476

Web Search History

date, time	web site	search string
5/4/2023 1:43:55 PM	Google Scholar	"distributed" "SDN" "app"
5/4/2023 1:44:03 PM	Google Scholar	"distributed" "SDN" "app" [before:2013]
5/4/2023 1:44:40 PM	Google Scholar	[before:2013] "distributed" "SDN" "app" "API"
5/4/2023 1:45:33 PM	Google Scholar	[before:2013] "distributed" "SDN" "app" "API" "programmable"
5/4/2023 1:46:44 PM	Google Scholar	[before:2013] "dSDN" "app" "API" "programmable" "flow"
5/4/2023 1:47:24 PM	Google Scholar	[before:2013] "SDN" "distributed applications" "API" "programmable" "flow"

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0477

Google Scholar "SDN" "distributed applications" "API" "programmable" "flow" SIGN IN

Articles About 89 results (0.05 sec) My profile My library

Any time
Since 2023
Since 2022
Since 2019
Custom range...
— 2013

Sort by relevance
Sort by date

Any type
Review articles

include patents
include citations

Create alert

An opennaas based sdn framework for dynamic qos control

I Bueno, JI Aznar, E Escalona, J Ferrer ... - 2013 IEEE SDN for ..., 2013 - ieeexplore.ieee.org

... of the inter-**SDN** routing information and the coordination of the **flow** setup across multiple ... The switch resource maps the OpenFlow rules into the controller through the REST **API** ...

Save Cite Cited by 83 Related articles All 2 versions

Software-Defined Networking and OpenFlow

S Das, D Talayco, R Sherwood - Handbook of Fiber Optic Data ..., 2013 - Elsevier

... that is above them that makes use of the network **API**. In ... **SDN** architecture decouples the control of data plane forwarding from the decision processes related to **distributed applications** ...

Save Cite Cited by 3 Related articles All 2 versions

[BOOK] Handbook of Fiber Optic Data Communication: Chapter 17. Software-Defined Networking and OpenFlow

S Das, D Talayco, R Sherwood - 2013 - books.google.com

... that is above them that makes use of the network **API**. In ... **SDN** architecture decouples the control of data plane forwarding from the decision processes related to **distributed applications** ...

Save Cite Cited by 1 Related articles

Is SDN the de-constraining constraint of the future internet?

[PDF] acm.org
Full Text @ USPTO-STIC

J Crowcroft, M Fidler, K Nahrstedt ... - ACM SIGCOMM Computer ..., 2013 - dl.acm.org

... **SDN** again would be the tool to replace the adhoc functionality with a clean slate **programmable** system with a public, open standard **API**... -scale, **SDN**, virtualization, and Open**Flow** seem ...

Save Cite Cited by 9 Related articles All 14 versions

[PDF] A compass through SDN networks

[PDF] tum.de

T Zinner, M Jarschel, T Hossfeld, P Tran-Gia ... - 2013 - mediatum.ub.tum.de

... via the Northbound-**API** between the **SDN** controller and what we ... technologies as well as **distributed applications**. This has led to ... , **SDN** allows load balancing to operate on any **flow** ...

Save Cite Cited by 5 Related articles All 3 versions

Design of a H. 261 video codec with 12 Xilinx LCAs

AM Praden, E Jouve, F Albert, M Dumas ... - Proceedings Euro ASIC ..., 1992 - computer.org

... readily available,to the **SDN** controller in achieving effective ... In-Memory Data Grid **distributed applications**,,in addition to ... - through its southbound OpenFlow **API**,to network view, and ...

Save Cite Related articles All 3 versions

Controller load balancing schemes in software defined networks

[PDF] yok.gov.tr

H Selvi - 2010 - acikbilim.yok.gov.tr

... on SimPy for **SDN** systems: We develop a **SDN** simulation ... It provides a standard OF **API** to the controller for control ... wide-area storage for **distributed applications**. HyperFlow is resilient ...

Save Cite

Towards Migrating Security Policies along with Virtual Machines in Cloud

[PDF] concordia.ca

S Sadri - 2013 - spectrum.library.concordia.ca

... of all **flow** entries contradicts an important **SDN** concept, ... In fact, as **distributed applications**, they are scattered around ... any **SDN** controller that offers a firewall module and a REST **API**, ...

Save Cite Cited by 2 Related articles All 2 versions

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0478

Cooperative routing between traffic control device and multi-server application [PDF] googleapis.com

L Deganaro, A Rodriguez - US Patent App. 11/224,691, 2007 - Google Patents

... routing requests through networks to **distributed applications**. ... TCP controls the **flow** of information on an end-to-end basis, ... **programmable** router is capable of accepting though its **API**. ...

Save Cite Cited by 6 Related articles All 2 versions

Open platform architecture for integrating multiple heterogeneous network functions [PDF] googleapis.com

K Fendick, L Kalampoukas, T Woo - US Patent 8,000,329, 2011 - Google Patents

... 3 , 4 and 5, it should be apparent to those skilled in the art that the logical path (ie, a **programmable** service path structure) that each packet **flow** takes when traversing through platform ...

Save Cite Cited by 123 Related articles All 4 versions

Main Concept Document
==========
US10893095
Title: Distributed software defined networking

(No Concept Modifiers)
==========

Filters (1)
==========
Publication Date: To 2013-12-31

Options
==========
De-dup: Extended Family Numbers (PAT only)
Relevance Cut-off: None
Sort: Relevance

Content: Patents + NPL (192)
==========
US Patents
US Designs
US Applications
EPO Patents
EPO Applications
China Patents
China Applications
Japan Patents
Japan Applications
Korea Patents
Korea Applications
WIPO Applications
Argentina Patents
Argentina Applications
Brazil Patents
Brazil Applications
Canada Patents
Canada Applications
Chile Patents
Chile Applications
Colombia Applications
Costa Rica Applications
Cuba Patents

Cuba Applications
Dominican Republic Applications
Ecuador Patents
Ecuador Applications
El Salvador Applications
Guatemala Applications
Honduras Applications
Mexico Patents
Mexico Applications
Nicaragua Patents
Panama Applications
Peru Applications
Trinidad & Tobago Patents
Uruguay Applications
Austria Patents
Austria Applications
Belarus Patents
Belgium Patents
Belgium Applications
Bosnia & Herzegovina Patents
Bosnia & Herzegovina Applications
Bulgaria Patents
Bulgaria Applications
Croatia Patents
Croatia Applications
Czech Republic Patents
Czech Republic Applications
Czechoslovakia Patents
Czechoslovakia Applications
Denmark Patents
Denmark Applications
Estonia Patents
Estonia Applications
EUIPO Patents
Finland Patents
Finland Applications
France Patents
France Applications
Germany Patents
Germany Applications
Great Britain Patents
Great Britain Applications
Greece Patents
Greece Applications
Hungary Patents
Hungary Applications
Iceland Patents
Iceland Applications

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0481

Ireland Patents
Ireland Applications
Italy Patents
Italy Applications
Latvia Patents
Latvia Applications
Lithuania Patents
Lithuania Applications
Luxembourg Patents
Luxembourg Applications
Malta Patents
Monaco Patents
Montenegro Patents
Montenegro Applications
Netherlands Patents
Netherlands Applications
Norway Patents
Norway Applications
Poland Patents
Poland Applications
Portugal Patents
Portugal Applications
Republic of Moldova Patents
Republic of Moldova Applications
Romania Patents
Romania Applications
San Marino Patents
San Marino Applications
Serbia Patents
Serbia Applications
Slovakia Patents
Slovakia Applications
Slovenia Patents
Spain Patents
Spain Applications
Sweden Patents
Sweden Applications
Switzerland Patents
Switzerland Applications
Ukraine Patents
Yugoslavia/Serbia and Montenegro Patents
Yugoslavia/Serbia and Montenegro Applications
Armenia Patents
Australia Patents
Australia Applications
Cyprus Patents
Gulf Cooperation Council Patents
Hong Kong Patents

India Patents
India Applications
Indonesia Patents
Indonesia Applications
Israel Patents
Israel Applications
Jordan Patents
Jordan Applications
Kyrgyzstan Patents
Macao Applications
Malaysia Patents
Mongolia Patents
New Zealand Patents
Philippines Patents
Philippines Applications
Saudi Arabia Patents
Saudi Arabia Applications
Singapore Patents
Singapore Applications
Taiwan Patents
Taiwan Applications
Tajikistan Patents
Tajikistan Applications
Thailand Patents
Thailand Applications
Uzbekistan Patents
Vietnam Patents
Algeria Patents
ARIPO Patents
ARIPO Applications
Egypt Patents
Kenya Patents
Malawi Patents
Morocco Patents
Morocco Applications
OAPI Patents
South Africa Patents
Tunisia Applications
Zambia Patents
Zimbabwe Patents
EAPO Patents
EAPO Applications
Georgia Patents
Georgia Applications
Kazakhstan Patents
Kazakhstan Applications
Russia Patents
Russia Applications

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0483

Turkey Patents
Turkey Applications
IEEE Xplore Publications: IEEE Periodicals
IEEE Xplore Publications: IEEE Conferences
IEEE Xplore Publications: IEEE Standards
IEEE Xplore Publications: IEEE Early Access
IEEE Xplore Publications: SMPTE Periodicals
IEEE Xplore Publications: SMPTE Conferences
IEEE Xplore Publications: SMPTE Standards
IEEE Xplore Publications: MIT Press eBooks
IEEE Xplore Publications: Wiley-IEEE eBooks
IEEE Xplore Publications: IBM Periodicals
IEEE Xplore Publications: URSI Periodicals
IEEE Xplore Publications: VDE Conferences
IEEE Xplore Publications: Periodicals from China
IP.com Prior Art Database: The IP.com Journal
IP.com Prior Art Database: Internet Society RFC
IP.com Prior Art Database: IBM TDB Archive
IP.com Prior Art Database: Legacy Journals
IP.com Prior Art Database: Software Patent Institute
OnePetro.org: Periodicals at OnePetro.org
OnePetro.org: Conferences at OnePetro.org
Other Literature: IBM Redbooks
Other Literature: PubMed Central
Other Literature: arXiv.org

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0484

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L1	30892	(H04L41/08,0895).cpc.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 11:17 AM
L2	169964	(H04L67/10).cpc.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 11:17 AM
L3	86593	(H04L45/563 H04L67/34).cpc.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 11:17 AM
L4	500	1 AND 2 AND 3	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU,	OR	ON	ON	2023/05/04 11:17 AM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0485

			CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L5	39	4 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 11:17 AM
L6	7	(docker kubernetes SDN NFV Open$1flow open$1stack) AND 5	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 11:19 AM
L9	613956	(docker kubernetes $1SDN NFV Open$1flow open$1stack)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN,	OR	ON	ON	2023/05/04 11:41 AM

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0486

			WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L10	22	1 AND 2 AND 9 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 11:42 AM
L11	172741	($2SDN NFV VNF $3app$1 $3appliance$2 VM cloud) adj10 (flow$4)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 11:57 AM
L12	2	1 AND 3 AND 11 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 11:58 AM
L13	11	1 AND 2 AND 11 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB,	OR	ON	ON	2023/05/04 12:04 PM

			HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L14	723847	($2SDN NFV VNF $3app$1 $3appliance$2 VM cloud) adj10 (pair$3 sister two couple match$3 team group$3)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 12:59 PM
L15	422	1 AND 14 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 01:00 PM
L16	27	15 AND (2 3)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT;	OR	ON	ON	2023/05/04 01:00 PM

			IBM_TDB)				
L17	322584	($2SDN NFV VNF $3app$1 $3appliance$2 VM cloud) adj10 (pair$3 sister coupl$3 match$3)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 01:01 PM
L18	335	1 AND 17 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/05/04 01:01 PM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N1	9932	(H04L41/08,0895).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/04 01:02 PM
N2	45245	(H04L67/10).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/04 01:02 PM
N3	21637	(H04L45/563 H04L67/34).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/04 01:02 PM
N4	297	1 AND 2 AND 3	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/04 01:02 PM
N5	15	N4 AND (@pd @ptad @prad @ad @rlad)<"20130613"	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/04 01:02 PM

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (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 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. **Because electronic patent issuance may occur shortly after issue fee payment, any desired continuing application should preferably be filed prior to payment of this issue fee in order not to jeopardize copendency.**

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

112449 7590 05/09/2023
Roark IP
1438 Dahlia Loop
San Jose, CA 95126

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 transmitted to the USPTO via EFS-Web or by facsimile to (571) 273-2885, on the date below.

(Typed or printed name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3	2943

TITLE OF INVENTION: Distributed Software Defined Networking

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0.00	$0.00	$480	08/09/2023

EXAMINER	ART UNIT	CLASS-SUBCLASS
HIGA, BRENDAN Y	2441	709-201000

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/AIA/122 or PTO/SB/122) attached.

❑ "Fee Address" indication (or "Fee Address" Indication form PTO/AIA/47 or 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 ROARK IP
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 must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: Edge Networking Systems, LLC

(B) RESIDENCE: (CITY and STATE OR COUNTRY): Austin, TX

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. Fees submitted: ☒ Issue Fee ❑ Publication Fee (if required)

4b. Method of Payment: *(Please first reapply any previously paid fee shown above)*

☒ Electronic Payment via Patent Center or EFS-Web ❑ Enclosed check ❑ Non-electronic payment by credit card (Attach form PTO-2038)

❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. _____

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 /Christopher J. Horgan/ Date May 16, 2023

Typed or printed name Christopher J. Horgan Registration No. 40,394

PTOL-85 Part B (08-18) Approved for use through 01/31/2020 OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0490

Electronic Patent Application Fee Transmittal

Application Number:	17142983
Filing Date:	06-Jan-2021
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Filer:	Christopher James Horgan
Attorney Docket Number:	TAAG001CONT3

Filed as Small Entity

Filing Fees for Utility under 35 USC 111(a)

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

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**480**

Electronic Acknowledgement Receipt	
EFS ID:	48012898
Application Number:	17142983
International Application Number:	
Confirmation Number:	2943
Title of Invention:	Distributed Software Defined Networking
First Named Inventor/Applicant Name:	Pouya Taaghol
Customer Number:	112449
Filer:	Christopher James Horgan
Filer Authorized By:	
Attorney Docket Number:	TAAG001CONT3
Receipt Date:	16-MAY-2023
Filing Date:	06-JAN-2021
Time Stamp:	17:53:50
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$480
RAM confirmation Number	E20235FH56108255
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	TAAG001CONT3_ISSUE_FEE_PAYMENT.pdf	142279 7ba6ff05aae0bdfbe20b01bdd2f5b38c8d974f36	no	1
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	38134 182e5c8e3bf71cc381646e6e9a192f7c932df9a5	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	180413		

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.

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.
17/142,983	07/04/2023	11695823	TAAG001CONT3	2943

112449 7590 06/14/2023

Roark IP
1438 Dahlia Loop
San Jose, CA 95126

ISSUE NOTIFICATION

The projected patent number and issue date are specified above. The patent will issue electronically. The electronically issued patent is the official patent grant pursuant to 35 U.S.C. § 153. The patent may be accessed on or after the issue date through Patent Center at https://patentcenter.uspto.gov/. The patent will be available in both the public and the private sides of Patent Center. Further assistance in electronically accessing the patent, or about Patent Center, is available by calling the Patent Electronic Business Center at 1-888-217-9197.

The USPTO is implementing electronic patent issuance with a transition period, during which period the USPTO will mail a ceremonial paper copy of the electronic patent grant to the correspondence address of record. Additional copies of the patent (i.e., certified and presentation copies) may be ordered for a fee from the USPTO's Certified Copy Center at https://certifiedcopycenter.uspto.gov/index.html. The Certified Copy Center may be reached at (800)972-6382.

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 Patents Stakeholder Experience (OPSE), Stakeholder Support Division (SSD) at (571)-272-4200.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0495

INVENTOR(s) (Please see PAIR WEB site http://pair.uspto.gov for additional inventors):

Pouya Taaghol, San Jose, CA;
Vivek Ramanna, San Jose, CA;

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Big Data Federation, Inc., Santa Clara, CA;

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.

Exhibit 1007
Microsoft Corporation v. Edge Networking Systems, LLC
0496

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.
17/142,983	01/06/2021	Pouya Taaghol	TAAG001CONT3	2943

112449 7590 07/06/2023

Roark IP
1438 Dahlia Loop
San Jose, CA 95126

EXAMINER	
HIGA, BRENDAN Y	

ART UNIT	PAPER NUMBER
2441	

MAIL DATE	DELIVERY MODE
07/06/2023	PAPER

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.

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.
17/142,983	04-Jul-2023	11695823

Roark IP
1438 Dahlia Loop
San Jose, CA 95126

EGRANT NOTIFICATION

Your electronic patent grant (eGrant) is now available, which can be accessed via Patent Center at https://patentcenter.uspto.gov

The electronic patent grant is the official patent grant under 35 U.S.C. 153. For more information, please visit https://www.uspto.gov/electronicgrants