NVO3
Internet-Draft
Intended status: Informational
Expires: April 21, 2016

Junying Pang
Jie Cao
Dapeng Liu
Dacheng Zhang
Alibaba
Yizhou Li
Hao Chen
Huawei Technologies
David Zhou
BoJian Wang
Deepak Kumar
Cisco Systems
Ruichang Gao
Yan Qiao
H3C

October 19 2015

Path Detection in VXLAN Overlay Network
draft-pang-nvo3-vxlan-path-detection-01

Abstract

In VXLAN overlay networks, Operation and Management(OAM)functions are important for fault management and performance monitoring. Path Detection(PD) is one critical OAM function which is applied to monitor and/or diagnose the potential paths between two VTEPs or between two Tenant System. In addition, it can assist to identify the locations of failures on data transmission paths.

This document specifies a method of PD method for VXLAN Overlay Networks by using a centralized controller. However,the method can be easily extended to support the overlay networks without a centrilized controller. It can also be generalized to other overlay technique such as NVGRE.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
  1.1. Acronyms and Terminology 4
2. Conventions Used in This Document 4
3. Path Detection Under the Assistance of Fabric Controller . . 4
  3.1. General Format of PD packet 4
  3.2. Format of VXLAN Header 5
  3.3. Pseudo-Header 6
  3.4. Format of OAM PDU 6
  3.5. Format of Extendable OAM TLV 7
    3.5.1. TLV Type 8
    3.5.2. Content of Extendable OAM TLV 8
4. Path Detection between VTEPs 10
5. Path Detection between Tenant Systems 12
6. Security Considerations 12
7. IANA Considerations 13
8. Acknowledgements 13
9. References 13
  9.1. Normative References 13
  9.2. Informative References 14
Authors' Addresses 14

1. Introduction

In VXLAN overlay networks, OAM functions such as fault management should be implemented to prevent the path failure problem [NVO3-OAM-REQ]. Path Detection is one of OAM function which can be used to detect all available paths between two Tenant Systems or two VTEPs, and so it is widely used to assist the identify of the failure locations along a transmission path.

In this memo, a PD mechanism is specified for VXLAN overlay networks. A centralized Controller is provided as a centralized unit to: 1) construct Path Detection(PD) packets, 2) inject them into the network devices to record information such as device's Ingress/Egress interface number, and 3) collect the PD packets from network devices for further analysis. Therefore, Path Detection such as monitoring and diagnose can be realized more efficient.

Figure 1 shows the architecture of this mechanism:

```
                 *******************************************
                 *              +------------+             *
                 *              |            |             *
                 *      +-------+   Fabric   +------+      *
                 *      |       | Controller |      |      *
                 *      |       |            |      |      *
                 *      |       +-----+------+      |      *
                 *      |             |             |      *
                 *      |             |             |      *
                 *      |             |             |      *
                 *      |             |             |      *
       +-----+   *  +---+--+     +----+----+     +--+---+  *   +-----+
       | +--+|   *  |+----+|     |         |     |+----+|  *   |+--+ |
       | |VM|+---*--+|VTEP|+-----+ L3      +-----+|VTEP|+--*---+|VM| |
       | +--+|   *  |+----+|     | Devices |     |+----+|  *   |+--+ |
       |     |   *  |      |     |         |     |      |  *   |     |
       +-----+   *  +------+     +---------+     +------+  *   +-----+
        Server   *    ToR                          ToR     *    Server
                 *                                         *
                 *                                         *
                 *         VXLAN Overlay Network           *
                 *                                         *
                 *******************************************
```

Figure 1: VXLAN Overlay Network with Fabric Controller

This method can be extended to support the overlay network without fabric controller. In this case, it could be regarded as a traditional OAM fault management solution described in draft-tissanvo3-oam-fm-02 [I-D.tissa-nvo3-oam-fm].It can also be generalized to support other overlay technique such as NVGRE [RFC7637].

The following of this document is organized as follows: Section 3 describes the format of PD packets. Section 4 introduces the procedure of Path Detection between VTEPs. Section 5 describes the procedure of Path Detection between Tenant Systems.

1.1. Acronyms and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Because this document reuses most of the terms specified in RFC 7348 [RFC7348] RFC 7364 [RFC7364] and RFC 7365 [RFC7365], this section only defines the key terms used by this document.

NVGRE: Network Virtualization using Generic Routing Encapsulation

OAM: Operations, Administration, and Management

Controller: an entity that generates PD packets and injects them into the overlay network through VTEPs, also collects PD packets from network devices in overlay network.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Path Detection Under the Assistance of Fabric Controller

This section describes the format of PD packet.

To provide accurate monitoring and/or diagnostic services, a PD packet and the corresponding user packets should be transported over the same data path. In addition, PD packets SHOULD NOT be transferred to the outside of the overlay network.

3.1. General Format of PD packet

Figure 2 shows the format of a PD packet:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
-      Outer MAC Header         -     14 Octets
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
-     Outer IPv4 Header         -     20 Octets
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Outer UDP Header         |      8 Octets
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         VXLAN Header          |      8 Octets
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
~        Pseudo-Header          ~    128 Octets
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            OAM PDU            |     Variable
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Format of the PD packet

VXLAN Header (8 Octets): A fixed size field, used to carry NVO3 specific information. This work complies with the VXLAN Header specified in Section 5 of [RFC 7348] but uses a reserve field as the flag to distinguish the packets for PD from the normal user packets.

Pseudo-Header (128 Octets): A fixed size field, consists of the information of Ethernet MAC header, IPv4 header, and TCP/UDP header, which is used to identify the packets within the same flow.

OAM PDU (Variable): A variable size field,used to carry the path detection information. An OAM PUB consists of OAM flag, OAM type and Extendable TLV as shown in Section 3.4. For a OAM PDU, 4 Octets alignment MUST be guaranteed.

3.2. Format of VXLAN Header

In this work, the "PD" flag(as with the illustration in Figure 3) MUST be set for all the PD packets.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+
|R|R|R|R|I|R|R|R|               Reserved                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        VXLAN Network Identifier (VNI)         |R|R|R|R|R|R|R|PD|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+
```

Figure 3: VXLAN header with the PD Flag

PD (1 bit) - Indicates it is a PD packet and needs to be handled as specified in this document.

All other fields comply with what are specified in Section 5 of RFC 7348 [RFC7348].

3.3. Pseudo-Header

The Pseudo-Header is used to ensure that the PD packets are transported along the paths that the service flows actually transported. In order to achieve this, the five-tuples identifying the service flow should be copied directly into associating fields in the Pseudo-header.

3.4. Format of OAM PDU

OAM PDU consists of an OAM flag field, an OAM type field and an Extendable TLV field. This structure is used to identify the type of Path Detection, and records the OAM information along the traverse path at each hop. The information will be report to fabric controller at each hop, in order to depict the complete path information. Following is the format of OAM PDU.

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   OAM Type    |               Reserved                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Extendable TLV (Variable)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Format of OAM PDU

OAM Type (1 Octet): used to identify the function of PD packets. Currently two functions are specified: path traversal and path tracking.

OAM type	Function
0x01	Path Traversal
0x02	Path Tracking
Other	Reserved

Reserved (3 Octets): padding bits, used to keep the 4 Octets alignment.

Extendable TLV (Variable): used to carry path detection information such as the Ingress/Egress Interface Identifiers of network devices along the path in VXLAN overlay network.

3.5. Format of Extendable OAM TLV

The following figure depicts the general format of an Extendable OAM TLV:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+
|             Type              |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Value (Variable)                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+
```

Figure 5: Extendable TLV of OAM PDU

Type (2 Octets): Specifies the Type of the TLV.(see Section 3.5.1 for TLV types)

Length (2 Octets): Specifies the length of the ’Value’ field in octets. Length of the ’field’ can be either zero or more octets.

Value (Variable): The length and the content of this field depend on the type of the TLV. (see Section 3.5.2 for content of TLV)

3.5.1. TLV Type

This document specifies two type of Extendable OAM TLV: Ingress Interface Identifier (IIID) TLV and Egress Interface Identifier (EIID) TLV. The Type field of each TLV is specified as follows:

Type	TLV Name
0x0001	Ingress Interface Identifier
0x0002	Egress Interface Identifier
0x0003	Transaction Identifier
0x0004	Ingress Interface Name Identifier
0x0005	Egress Interface Name Identifier
0x0006	Authentication

3.5.2. Content of Extendable OAM TLV

For an IIID TLV, the type field is set as 0x0001, the length field is set as 4. The value field is 4 Octets long which contains the device’s Ingress Interface Identifier.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++--+-+-+
|            Type = 0x0001      |           Length = 4          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Ingress Interface Identifier                      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: IIID TLV

For an EIID TLV, the type field is set as 0x0002, the length field is set as 4. The value field is 4 Octets long which contains the device’s Egress Interface Identifier.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = 0x0002        |           Length = 4          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Egress Interface Identifier                      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: EIID TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = 0x0003        |           Length = 16         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Transaction  Identifier                    |
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Transaction Identifier TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = 0x0004        | Length = ifName length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                ifName(RFC2863 (DisplayString))                |
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Ingress Interface Name Identifier TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = 0x0005        | Length = ifName length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                ifName(RFC2863 (DisplayString))                |
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Egress Interface Name Identifier TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = 0x0006        |        Length = 20            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Key Id       |                      Reserved                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
|                             Key                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Key ID: 8 bits. This allows multiple keys to be active simultaneously.

Auth Key: 16 octets. This field carries the MD5 [RFC1321] checksum for the entire IP packet. When the Auth Key is calculated, the shared MD5 key is stored in this field, and the checksum fields in the IP header, UDP header are set to zero
.

The result of the algorithm is placed in the Key field.

Figure 11: Authentication TLV

4. Path Detection between VTEPs

In VXLAN overlay networks, Equal Cost Multi-path(ECMP) may exist between two VTEPs, which may be leveraged to achieve load balance. Link failure and other reasons may lead to the broken of equal cost paths. In order to avoid delivering packets to the broken paths, it’s necessary to detect all the potential paths between the VTEPs. The basic idea is to traversal these paths using the PD packets.

The process of path detection between VTEPs is:

1. The Fabric Controller generates a series of Path Traversal packets targeting to the same Egress VTEP. The outer Source UDP port numbers of the Path Traversal packets keep increased by 1. For example, assume the outer Source UDP port number of a Path Traversal packet is set to 4000. Then the outer Source UDP port numbers of subsequent Path Traversal packets are set as 4001, 4002, 4003, etc. The 'PD bit' in VXLAN header is set to 1 . The content of Pseudo header is left to empty (default value is full-zero), and the OAM type field is set to 0x01 to indicate that it is a Path Traversal packet.

2. After the Ingress VTEP receives the Path Traversal packets from Controller, it then computes the corresponding egress port based on the outer header information and then delivers the packet to that port. By continuous increasing the Source UDP port number, these packets can be distributed to different equal cost paths. Therefore, these Path Traversal packets could go through all the equal cost paths between the two VTEPs.

3. The Extendable TLV field contains multipls TLVs. Transaction Identifier TLV is set by controller and carried in packet without modification in scenario where multiple transactions are initiated by controller between two endpoints. Network device can add Ingress Interface Identifier or Ingress Interface Name Identifier, and Egress Interface Identifier or Egress Interface Name Identifier starting at the end of Transaction Identifier TLV. Both of these TLVs are set by the network devices along the transport path. The TLVs are used to record the identifier of device's Ingress/Egress interface the PD packet goes through. Each network device receives the Path Traversal packet from its upstream device, makes a copy of it and passes the copy to its CPU. After filling the extendable TLVs in this copy, the network device will deliver this copy to the Fabric Controller for further handling.

4. As new TLVs are added by network device in payload section of UDP/Ipv4 packet, it's good practise to update the IP length, UDP length and IP CRC.

5. By gathering all the Path traversal packets from the network devices along the paths, the Controller is able to compute the number of available paths, which could be presented by graphical chart.

5. Path Detection between Tenant Systems

In VXLAN overlay network, link failures are common and it may affect normal operations of up-layer applications. For example, it may lead to service flow interruptions which are unacceptable for most applications.

Path Detection between two End Systems is essential for accurate monitoring and/or diagnostics. The basic idea is to transport the Path Tracking packets right along the path, that the service flow are transport through.

The process of Path Detection between Tenant Systems is:

1. The Fabric Controller generates one Path Tracking packet to Ingress VTEP. The 'PD bit' in the VXLAN header of the packet is set to 1. The content of Pseudo-header is set as the tuple information which are transported over the path being detected. The OAM type field is set to 0x02 to indicate it is a Path Tracking packet.

2. After the Ingress VTEP receives the Path Tracking packets from Fabric Controller, it will firstly compute the outer source UDP port number based on the information form in Pseudo-header. Then it deliveries these packets to the corresponding egress port based on the outer headers information.

3. Each network device receives the Path Tracking packet from its upstream device, makes a copy of it and passes the copy to its CPU. After filling the extendable TLVs in this copy, the network device will deliver this copy to the Controller for further handling. By doing this, each network device along the path will deliver a copy of Path Tracking packets back to Fabric Controller in a hop-by-hop manner.

4. By gathering all the Path traversal packets from network devices along the paths, Fabric Controller is able to accurately monitor the status of each link on the data flow path and locate the point of failure. Fabric Controller may also present the path status using graphical chart.

6. Security Considerations

VXLAN security consideration is discussed in Section 7 of RFC 7348. This document specifies a path failure detection mechanism by extending the VXLAN header. Thus it has the similar vulnerability as VXLAN. For example, attackers can inject spoofed path failure detection packets to the VXLAN overlay network. Administrative measures, ACL(Access Control List), authentication and encryption etc could be used to mitigate the attack.

In addition, because the controller needs to collect and process the PD packets sent from the network devices. An attacker may perform DDoS attacks to the controller by generating a large amount of PD packets and sent them to a VXLAN overlay network. This issue will be well analyzed in our future work.

As communication between controller and network switch is over internet and it’s IP traffic, IPSEC Encryption [RFC 6071] may be used to encrypt the communication.

7. IANA Considerations

TBD.

8. Acknowledgements

The authors would like to specially thank Daolong zhou for his generous help in improving the readability of this document.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <http://www.rfc-editor.org/info/rfc7348>.

[RFC7364] Narten, T., Ed., Gray, E., Ed., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", RFC 7364, DOI 10.17487/RFC7364, October 2014, <http://www.rfc-editor.org/info/rfc7364>.

9.2. Informative References

[I-D.ashwood-nvo3-oam-requirements]
Chen, H., Ashwood-Smith, P., Xia, L., Iyengar, R., Tsou, T., Sajassi, A., Boucadair, M., Jacquenet, C., Daikoku, M., Ghanwani, A., and R. Krishnan, "NVO3 Operations, Administration, and Maintenance Requirements", draft-ashwood-nvo3-oam-requirements-03 (work in progress), July 2015.

[I-D.tissa-nvo3-oam-fm]
Senevirathne, T., Salam, S., Kumar, D., Finn, N., Eastlake, D., and S. Aldrin, "NVO3 Fault Management", draft-tissa-nvo3-oam-fm-02 (work in progress), June 2015.

[RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <http://www.rfc-editor.org/info/rfc7365>.

[RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <http://www.rfc-editor.org/info/rfc7637>.

Authors' Addresses

Junying Pang
Alibaba

Email: kittypang@alibaba-inc.com

Jie Cao
Alibaba

Email: jie.caojie@alibaba-inc.com

Dapeng Liu
Alibaba

Email: max.ldp@alibaba-inc.com

Dacheng Zhang
Alibaba

Email: dacheng.zdc@alibaba-inc.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56624440
Email: liyizhou@huawei.com

Hao Chen
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56624440
Email: philips.chenhao@huawei.com

David Zhou
Cisco Systems
China

BoJian Wang
Cisco Systems
China

Deepak Kumar
Cisco Systems
USA

Email: dekumar@cisco.com

Ruichang Gao
H3C
China

Email: gaoruichang@h3c.com

Yan Qiao
H3C
China

Email: qiaoyan@h3c.com