                    The IPv6 Compressed Routing Header (CRH)
                       draft-bonica-6man-comp-rtg-hdr-06

Abstract

   This document defines two new IPv6 Routing header types.
   Generically, they are called the Compressed Routing Header (CRH).
   More specifically, the 16-bit version of the CRH is called the CRH-
   16, while the 32-bit version of the CRH is called the CRH-32. SRv6+
   nodes use the CRH to steer packets from segment to segment along
   SRv6+ paths.

Status of This Memo

   This Internet-Draft will expire on February 29, 2020.

Copyright Notice

Table of Contents

1. Introduction

This document defines two new IPv6 [RFC8200] Routing header types.
Generically, they are called the Compressed Routing Header (CRH).
More specifically, the 16-bit version of the CRH is called the CRH-
16, while the 32-bit version of the CRH is called the CRH-32. SRv6+
[I-D.bonica-spring-srv6-plus] nodes use the CRH to steer packets from
segment to segment along SRv6+ paths.

For details regarding SRv6+ paths, segments, Segment Identifiers
(SIDs) and instructions, see [I-D.bonica-spring-srv6-plus].

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in BCP
14 [RFC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here.

3. The Compressed Routing Header (CRH)

Both CRH versions (i.e., CRH-16 and CRH-32) contain the following
fields:

o Next Header - Defined in [RFC8200]. Identifies the type of header
   immediately following the CRH.

o Hdr Ext Len - Defined in [RFC8200]. Length of the CRH in 8-octet
   units, not including the first 8 octets.

o Routing Type - Defined in [RFC8200]. Value TBD by IANA. (For
   CRH-16, the suggested value is 5. For CRH-32, the suggested value
   is 6.)

o Segments Left - Defined in [RFC8200]. Number of route segments
   remaining, i.e., number of explicitly listed intermediate nodes
   still to be visited before reaching the final destination.

o SID List - Represents the SRv6+ path as an ordered list of SIDs.
   SIDs are listed in reverse order, with SID[0] representing the
   final segment, SID[1] representing the penultimate segment, and so
   forth. SIDs are listed in reverse order so that Segments Left can
   be used as an index to the SID List. The SID indexed by Segments
   Left is called the current SID.

In the CRH-16 (Figure 1), each SID list entry is encoded in 16-bits.
In the CRH-32 (Figure 2), each SID list entry is encoded in 32-bits.

In networks where the smallest feasible Maximum SID Value (MSV)
[I-D.bonica-spring-srv6-plus] is greater than 65,635, CRH-32 is
required. Otherwise, CRH-16 is preferred.

In all cases, the CRH MUST end on a 64-bit boundary. Therefore, the
CRH MAY be padded with zeros.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Next Header  |  Hdr Ext Len  |  Routing Type | Segments Left |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            SID[0]             |            SID[1]             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
   |                         .........
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```


                         Figure 1: CRH-16

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Next Header  |  Hdr Ext Len  | Routing Type  | Segments Left |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   +                            SID[0]                             +
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   +                            SID[1]                             +
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                                                             //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   +                            SID[n]                             +
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
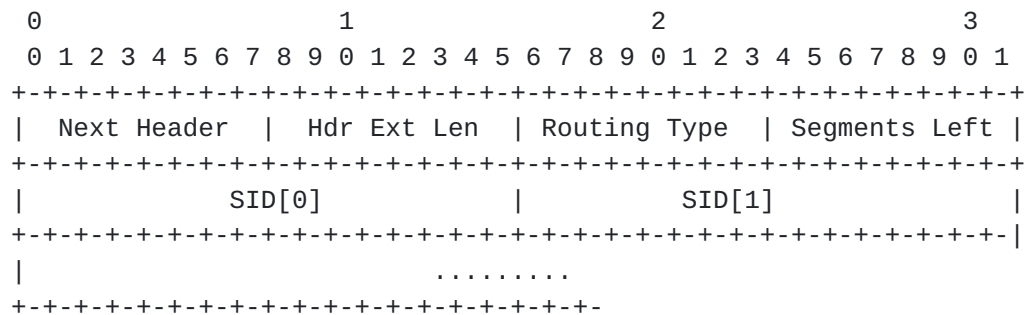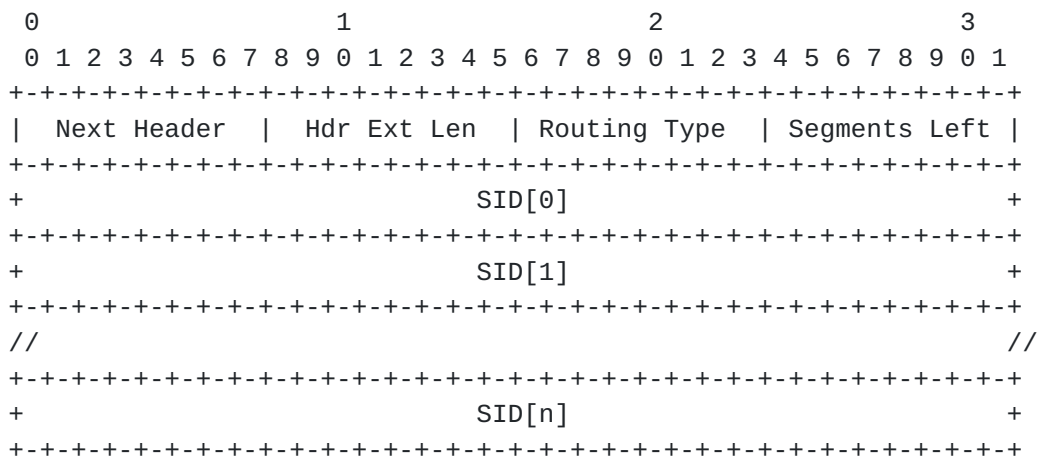

                         Figure 2: CRH-32

4. The Segment Forwarding Information Base (SFIB)

A segment ingress node MUST maintain one Segment Forwarding
Information Base (SFIB) entry for each segment that it originates.
Each SFIB entry contains the following information:

o A SID.

o A segment type.

o Topological instruction parameters.

The following are valid segment types:

o Adjacency.

o Node.

The following parameters are associated with topological instructions
that control adjacency segments:

o An IPv6 address that identifies an interface on the segment egress
   node.

o An interface identifier.

Node segments are associated with a single topological instruction
parameter. This parameter is an IPv6 address that identifies an
interface on the segment egress node.

5. Processing Rules

5.1. General

[RFC8200] defines rules that apply to IPv6 extension headers, in
general, and IPv6 Routing headers, in particular. All of these rules
apply to the CRH.

For example:

o Extension headers (except for the Hop-by-Hop Options header) are
   not processed, inserted, or deleted by any node along a packet's
   delivery path, until the packet reaches the node (or each of the
   set of nodes, in the case of multicast) identified in the
   Destination Address field of the IPv6 header.

o If, while processing a received packet, a node encounters a
   Routing header with an unrecognized Routing Type value, the
   required behavior of the node depends on the value of the Segments
   Left field. If Segments Left is zero, the node must ignore the
   Routing header and proceed to process the next header in the
   packet, whose type is identified by the Next Header field in the
   Routing header. If Segments Left is non-zero, the node must
   discard the packet and send an ICMPv6 [RFC4443] Parameter Problem,
   Code 0, message to the packet's Source Address, pointing to the
   unrecognized Routing Type.

o If, after processing a Routing header of a received packet, an
   intermediate node determines that the packet is to be forwarded
   onto a link whose link MTU is less than the size of the packet,
   the node must discard the packet and send an ICMPv6 Packet Too Big
   message to the packet's Source Address.

5.2. CRH Specific

When a node recognizes and processes a CRH, it executes the following
procedure:

o If the IPv6 Source Address is a link-local address, discard the
   packet.

o If the IPv6 Source Address is a multicast address, discard the
   packet.

o If Segments Left equals 0, skip over the CRH and process the next
   header in the packet.

o If Hdr Ext Len indicates that the CRH is larger than the
   implementation can process, discard the packet and send an ICMPv6
   Parameter Problem, Code 0, message to the Source Address, pointing
   to the Hdr Ext Len field.

o Compute L, the minimum CRH length (See Section 5.2.1).

o If L is greater than Hdr Ext Len, discard the packet and send an
   ICMPv6 Parameter Problem, Code 0, message to the Source Address,
   pointing to the Segments Left field.

o Decrement the packet's Hop Count.

o If the Hop Count has expired, discard the packet and send an
   ICMPv6 Time Expired message to the packet's source node.

o Decrement Segments Left

o Search for the current SID in the SFIB.

o If the above-mentioned search does not return an SFIB entry,
   discard the packet and send an ICMPv6 Parameter Problem, Code 0,
   message to the Source Address, pointing to the current SID.

o If the above-mentioned search returns an SFIB entry that
   represents an adjacency segment, execute the topological
   instruction described in Section 5.2.2.

o If the above-mentioned search returns an SFIB entry that
   represents a node segment, execute the topological instruction
   described in Section 5.2.3.

The above stated rules are demonstrated in Appendix A.

5.2.1. Computing Minimum CRH Length

The algorithm described in this section accepts the following CRH
fields as its input parameters:

o Routing Type (i.e., CRH-16 or CRH-32).

o Segments Left.

It yields L, the minimum CRH length. The minimum CRH length is
measured in 8-octet units, not including the first 8 octets.

```
<CODE BEGINS>

switch(Routing Type) {
    case CRH-16:
        sidsBeyondFirstWord = Segments Left - 2;
        sidPerWord = 4;
    case CRH-32:
        sidsBeyondFirstWord = Segments Left - 1;
        sdsPerWord = 2;
    case default:
        return(0xFF);
    }

if (sidsBeyondFirstWord <= 0)
    return(0)

words = sidsBeyondFirstWord div sidsPerWord;
if (sidsBeyondFirstWord mod sidsPerWord)
    words++;

return(words)


<CODE ENDS>
```

5.2.2. Topological Instructions That Control Adjacency Segments

   A topological instruction that controls an adjacency segment accepts
   the following parameters:

   o An IPv6 address that identifies an interface on the segment egress
      node.

   o An interface identifier.

   The instruction behaves as follows:

   o If the interface that was received as a parameter is not
      operational, discard the packet and send an ICMPv6 Destination
      Unreachable message (Code: 5, Source Route Failed) to the packet's
      source node.

   o Overwrite the packet's Destination Address with the IPv6 address
      that was received as a parameter.

   o Forward the packet through the above-mentioned interface.

5.2.3. Topological Instructions That Control Node Segments

   A topological instruction that controls a node segment accepts a
   single parameter. This parameter is an IPv6 address that identifies
   an interface on the segment egress node.

   The instruction behaves as follows:

   o If the segment ingress node does not have a viable route to the
      IPv6 address included as a parameter, discard the packet and send
      an ICMPv6 Destination Unreachable message (Code:1 Net Unreachable)
      to the packet's source node.

   o Overwrite the packet's Destination Address with the destination
      address that was included as a parameter.

   o Forward the packet to the next hop along the least cost path to
      the segment egress node. If there are multiple least cost paths
      to the segment egress node (i.e., Equal Cost Multipath), execute
      procedures so that all packets belonging to a flow are forwarded
      through the same next hop.

6. Mutability

In the CRH, the Segments Left field is mutable. All remaining fields are immutable.

7. Compliance

In order to be compliant with this specification, an SRv6+ implementation MUST:

o Be able to process IPv6 options as described in Section 4.2 of [RFC8200].

o Be able to process the Routing header as described in Section 4.4 of [RFC8200].

o Support the CRH-16 and the CRH-32.

8. Management Considerations

PING and TRACEROUTE [RFC2151] both operate correctly in the presence of the CRH.

9. Security Considerations

SRv6+ domains MUST NOT span security domains. In order to enforce this requirement, security domain edge routers MUST do one of the following:

o Discard all inbound SRv6+ packets whose IPv6 destination address represents domain infrastructure.

o Authenticate [RFC4302] [RFC4303] all inbound SRv6+ packets whose IPv6 destination address represents domain infrastructure.

10. IANA Considerations

IANA is requested to make the following entries in the Internet Protocol Version 6 (IPv6) Parameters "Routing Type" registry:

```
   Suggested
   Value           Description                           Reference
   --------------------------------------------------------------------
     5       Compressed Routing Header (16-bit) (CRH-16)   This document
     6       Compressed Routing Header (32-bit) (CRH-32)   This document
```

11. Acknowledgements

Thanks to Joel Halpern, Tony Li, Gerald Schmidt, Nancy Shaw, and
Chandra Venkatraman for their comments.

12. References

12.1. Normative References

[I-D.bonica-spring-srv6-plus]
          Bonica, R., Hegde, S., Kamite, Y., Alston, A., Henriques,
          D., Halpern, J., Linkova, J., and G. Chen, "IPv6 Support
          for Segment Routing: SRv6+", draft-bonica-spring-
          srv6-plus-04 (work in progress), July 2019.

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

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302,
          DOI 10.17487/RFC4302, December 2005,
          <https://www.rfc-editor.org/info/rfc4302>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)",
          RFC 4303, DOI 10.17487/RFC4303, December 2005,
          <https://www.rfc-editor.org/info/rfc4303>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet
          Control Message Protocol (ICMPv6) for the Internet
          Protocol Version 6 (IPv6) Specification", STD 89,
          RFC 4443, DOI 10.17487/RFC4443, March 2006,
          <https://www.rfc-editor.org/info/rfc4443>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
          2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
          May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6
          (IPv6) Specification", STD 86, RFC 8200,
          DOI 10.17487/RFC8200, July 2017,
          <https://www.rfc-editor.org/info/rfc8200>.

12.2. Informative References

   [RFC2151]   Kessler, G. and S. Shepard, "A Primer On Internet and TCP/
               IP Tools and Utilities", FYI 30, RFC 2151,
               DOI 10.17487/RFC2151, June 1997,
               <https://www.rfc-editor.org/info/rfc2151>.

Appendix A. CRH Processing Examples

   This appendix demonstrates CRH processing in the following scenarios:

   o   SR path contains node segments only (Appendix A.1).

   o   SR path contains node segments only and preserves the first SID
       (Appendix A.2).

   o   SR path contains adjacency segments only (Appendix A.3).

```
                             -----------
            2001:db8:0:2/64   |Node: I2   |   2001:db8:0:4/64
           ---------------------|Loopback:  |--------------------
           |                ::2 |2001:db8::2| ::1                |
           |                    -----------                      |
           | ::1                                            :: 2|
     -----------              -----------              -----------
    |Node: S    |2001:db8:0:1/64|Node: I1   |2001:db8:0:3/64|Node: I3    |
    |Loopback   |---------------|Loopback:  |---------------|Loopback:   |
    |2001:db8::a| ::1       ::2 |2001:db8::1| ::1       ::2 |2001:db8::3|
     -----------              -----------              -----------
                                                            | ::1
                             -----------                    |
                             |Node: D    |   2001:db8:0:b/64   |
                             |Loopback:  |--------------------
                             |2001:db8::b| ::2
                             -----------
```
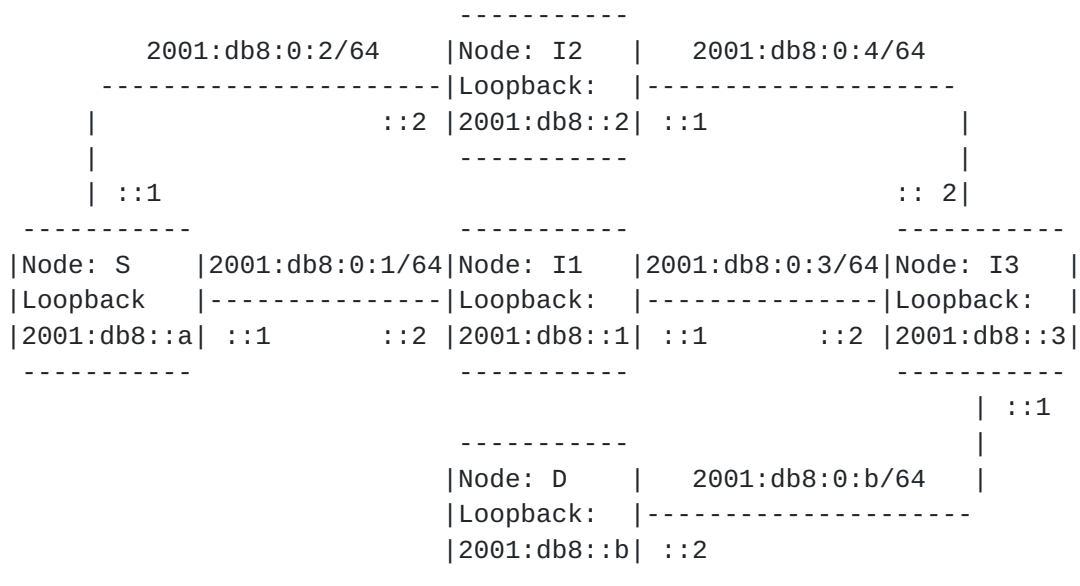
                    Figure 3: Reference Topology

   Figure 3 provides a reference topology that is used in all examples.

```
+--------------------+-----+-------------+-------------+
| Instantiating Node | SID | Segment Type | IPv6 Address |
+--------------------+-----+-------------+-------------+
|        All         | 1   | Node        | 2001:db8::1 |
|        All         | 2   | Node        | 2001:db8::2 |
|        All         | 3   | Node        | 2001:db8::3 |
|        All         | 10  | Node        | 2001:db8::a |
|        All         | 11  | Node        | 2001:db8::b |
+--------------------+-----+-------------+-------------+
```

                    Table 1: Node SIDs

   Table 1 describes SFIB entries that are instantiated on all nodes.
   All of these SFIB entries represent node segments.

```
+--------------------+-----+----------------+-----------+
| Instantiating Node | SID |   IPv6 Address  | Interface |
+--------------------+-----+----------------+-----------+
|         S          | 129 | 2001:db8:0:1::2 |  S -> I1  |
|         S          | 130 | 2001:db8:0:2::2 |  S -> I2  |
|         I1         | 129 | 2001:db8:0:3::2 |  I1 -> I3 |
|         I2         | 129 | 2001:db8:0:4::2 |  I2 -> I3 |
|         I3         | 129 | 2001:db8:0:b::2 |  I3 -> D  |
+--------------------+-----+----------------+-----------+
```

                    Table 2: Adjacency SIDs

   Table 2 describes SFIB entries that are instantiated on specific
   nodes. All of these SFIB entries represent adjacency segments.

A.1. SR Path Contains Node Segments Only

   In this example, Node S sends a packet to Node D, though a node
   segment that terminates on I3. In this example, I3 does not appear
   in the CRH segment list. Therefore, the destination node may not be
   able to send return traffic through the same path.

```
+------------------------------------+------------------+
| As the packet travels from S to I3: |                  |
+------------------------------------+------------------+
| Source Address = 2001:db8::a       | Last Entry = 0    |
| Destination Address = 2001:db8::3  | Segments Left = 1 |
|                                    | SID[0] = 11       |
+------------------------------------+------------------+
```

```
+-------------------------------------+------------------+
| As the packet travels from I3 to D: |                  |
+-------------------------------------+------------------+
| Source Address = 2001:db8::a        | Last Entry = 0   |
| Destination Address = 2001:db8::b   | Segments Left = 0 |
|                                     | SID[0] = 11      |
+-------------------------------------+------------------+
```

A.2. **SR Path Contains Node Segments Only And Preserves The First SID**

In this example, Node S sends a packet to Node D, through a node
segment that terminates on I3. In this example, I3 appears in the
CRH segment list. Therefore, the destination node can send return
traffic through the same path.

```
+-------------------------------------+------------------+
| As the packet travels from S to I3: |                  |
+-------------------------------------+------------------+
| Source Address = 2001:db8::a        | Last Entry = 1   |
| Destination Address = 2001:db8::3   | Segments Left = 1 |
|                                     | SID[0] = 11      |
|                                     | SID[1] = 3       |
+-------------------------------------+------------------+


+-------------------------------------+------------------+
| As the packet travels from I3 to D: |                  |
+-------------------------------------+------------------+
| Source Address = 2001:db8::a        | Last Entry = 1   |
| Destination Address = 2001:db8::b   | Segments Left = 0 |
|                                     | SID[0] = 11      |
|                                     | SID[1] = 3       |
+-------------------------------------+------------------+
```

A.3. **SR Path Contains Adjacency Segments Only**

In this example, Node S sends a packet to Node D, via two adjacency
segments..

```
+---------------------------------------+------------------+
| As the packet travels from S to I1:   |                  |
+---------------------------------------+------------------+
| Source Address = 2001:db8::a          | Last Entry = 1   |
| Destination Address = 2001:db8:0:1::2 | Segments Left = 2 |
|                                       | SID[0] = 129     |
|                                       | SID[1] = 129     |
+---------------------------------------+------------------+
```

```
+---------------------------------------+------------------+
| As the packet travels from I1 to I3:  |                  |
+---------------------------------------+------------------+
| Source Address = 2001:db8::a          | Last Entry = 1   |
| Destination Address = 2001:db8:0:3::2 | Segments Left = 1 |
|                                       | SID[0] = 129     |
|                                       | SID[1] = 129     |
+---------------------------------------+------------------+


+---------------------------------------+------------------+
| As the packet travels from I3 to D:   |                  |
+---------------------------------------+------------------+
| Source Address = 2001:db8::a          | Last Entry = 1   |
| Destination Address = 2001:db8:0:b::2 | Segments Left = 0 |
|                                       | SID[0] = 129     |
|                                       | SID[1] = 129     |
+---------------------------------------+------------------+
```

Authors' Addresses

Ron Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, Virginia 20171
USA

Email: rbonica@juniper.net

Yuji Kamite
NTT Communications Corporation
3-4-1 Shibaura, Minato-ku
Tokyo 108-8118
Japan

Email: : y.kamite@ntt.com

Tomonobu Niwa
KDDI
3-22-7, Yoyogi, Shibuya-ku
Tokyo 151-0053
Japan

Email: to-niwa@kddi.com

Andrew Alston
Liquid Telecom
Nairobi
Kenya

Email: Andrew.Alston@liquidtelecom.com

Daniam Henriques
Liquid Telecom
Johannesburg
South Africa

Email: daniam.henriques@liquidtelecom.com

Ning So
Reliance Jio
3010 Gaylord PKWY, Suite 150
Frisco, Texas 75034
USA

Email: Ning.So@ril.com

Fengman Xu
Reliance Jio
3010 Gaylord PKWY, Suite 150
Frisco, Texas 75034
USA

Email: Fengman.Xu@ril.com

Gang Chen
Baidu
No.10 Xibeiwang East Road Haidian District
Beijing 100193
P.R. China

Email: phdgang@gmail.com

Yongqing Zhu
China Telecom
109 West Zhongshan Ave, Tianhe District
Guangzhou
P.R. China

Email: zhuyq.gd@chinatelecom.cn

Guangming Yang
China Telecom
109 West Zhongshan Ave, Tianhe District
Guangzhou
P.R. China

Email: yanggm.gd@chinatelecom.cn

Yifeng Zhou
ByteDance
Building 1, AVIC Plaza, 43 N 3rd Ring W Rd Haidian District
Beijing 100000
P.R. China

Email: yifeng.zhou@bytedance.com