        Directory Assistance Problem and High-Level Design Proposal

Abstract

   Edge TRILL (Transparent Interconnection of Lots of Links) switches
   currently learn the mapping between MAC (Media Access Control)
   addresses and their egress TRILL switch by observing the data packets
   they ingress or egress or by the TRILL ESADI (End-Station Address
   Distribution Information) protocol. When an ingress TRILL switch
   receives a data frame for a destination address (MAC&Label) that the
   switch does not know, the data frame is flooded within the frame's
   Data Label across the TRILL campus.

   This document describes the framework for using directory services to
   assist edge TRILL switches in reducing multi-destination frames,
   particularly unknown unicast frames flooding, and ARP/ND (Address
   Resolution Protocol / Neighbor Discovery), thus improving TRILL
   network scalability and security.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   Edge TRILL (Transparent Interconnection of Lots of Links) switches
   (devices implementing [RFC6325], also known as RBridges) currently
   learn the mapping between destination MAC addresses and their egress
   TRILL switch by observing data packets or by the ESADI (End-Station
   Address Distribution Information) protocol. When an ingress RBridge
   (Routing Bridge) receives a data frame for a destination address
   (MAC&Label) that RBridge does not know, the data frame is flooded
   within that Data Label across the TRILL campus. (Data Labels are
   VLANs or FGLs (Fine-Grained Labels [FGL]).

   This document describes a framework for using directory services in
   environments where such services are available, such as typical data
   centers, to assist edge TRILL switches. This assistance can reduce
   multi-destination frames, particularly ARP [RFC826], ND [RFC4861],
   and unknown unicast, thus improving TRILL network scalability. In
   addition, the information provided by a directory can be more secure
   than that learned from the data plane (see Section 7).

   Data centers, especially Internet and/or multi-tenant data centers,
   tend to have a large number of end stations with a wide variety of
   applications. Their networks differ from enterprise campus networks
   in several ways that make them attractive for the use of directory
   assistance, in particular:

   1. Data center topology is often based on racks and rows.
      Furthermore, a Server/VM (virtual machine) Management System
      orchestrates the assignment of guest operating systems to servers,
      racks, and rows; it is not done at random. So, the information
      necessary for a directory is normally available from that
      Management System.

   2. Rapid workload shifting in data centers can accelerate the
      frequency of the physical servers being reloaded with different
      applications. Sometimes, applications loaded into one physical
      server at different times can belong to different subnets. When a
      VM is moved to a new location or when a server is loaded with a
      new application with different IP/MAC addresses, it is more likely
      that the destination address of data packets sent out from those
      VMs are unknown to their attached edge RBridges.

   3. With server virtualization, there is an increasing trend to
      dynamically create or delete VMs when the demand for resources
      changes, to move VMs from overloaded servers to less loaded
      servers, or to aggregate VMs onto fewer servers when demand is
      light. This results in the more frequent occurrence of multiple

        subnets on the same port at the same time and a higher change rate
        for VMs than for physical servers.

   Both items 2 and 3 above can lead to applications in one subnet being
   placed in different locations (racks or rows) or one rack having
   applications belonging to different subnets.

2. Terminology

   The terms "VLAN" and "Data Label" are used interchangeably with
   "Subnet" in this document, because it is common to map one subnet to
   one VLAN or FGL.

   Bridge: Device compliant with IEEE Std 802.1Q-2011 [802.1Q].

   Data Label: VLAN or FGL

   EoR: End-of-Row switches in a data center. Also known as
            aggregation switches.

   End Station: Guest OS running on a physical server or on a virtual
            machine. An end station in this document has at least one
            IP address, at least one MAC address, and is connected to a
            network.

   FGL: Fine-Grained Label [FGL]

   IS-IS: Intermediate System to Intermediate System routing protocol.
            TRILL uses IS-IS [IS-IS] [RFC6326].

   RBridge: "Routing Bridge", an alternate name for a TRILL switch.

   ToR: Top-of-Rack switches in a data center. Also known as access
            switches in some data centers.

   TRILL: Transparent Interconnection of Lots of Links [RFC6325]

   TRILL Switch: A device implementing the TRILL protocol [RFC6325].

   VM: Virtual Machine

3. Impact of Massive Number of End Stations

   This section discusses the impact of a massive number of end stations
   in a TRILL campus using Data Centers as an example.

3.1. Issues of Flooding-Based Learning in Data Centers

   It is common for Data Center networks to have multiple tiers of
   switches, for example, one or two Access Switches for each server
   rack (ToR), aggregation switches for some rows (or EoR switches), and
   some core switches to interconnect the aggregation switches. Many
   aggregation switches deployed in data centers have high port density.
   It is not uncommon to see aggregation switches interconnecting
   hundreds of ToR switches.

```
                    +-------+          +------+
                  +/------+ |        +/-----+ |
                  | Aggr11| + ----- |AggrN1| +    EoR switches
                  +---+---+/        +------+/
                   /     \           /      \
                  /       \         /        \
             +---+     +---+     +---+      +---+
             |T11|... |T1x|     |T21| ..  |T2y| ToR switches
             +---+     +---+     +---+      +---+
              |         |         |         |
             +-|-+     +-|-+     +-|-+     +-|-+
             |   |... |   |     |   | .. |   |
             +---+     +---+     +---+      +---+ Server racks
             |   |... |   |     |   | .. |   |
             +---+     +---+     +---+     +---+
             |   |... |   |     |   | .. |   |
             +---+     +---+     +---+     +---+
```
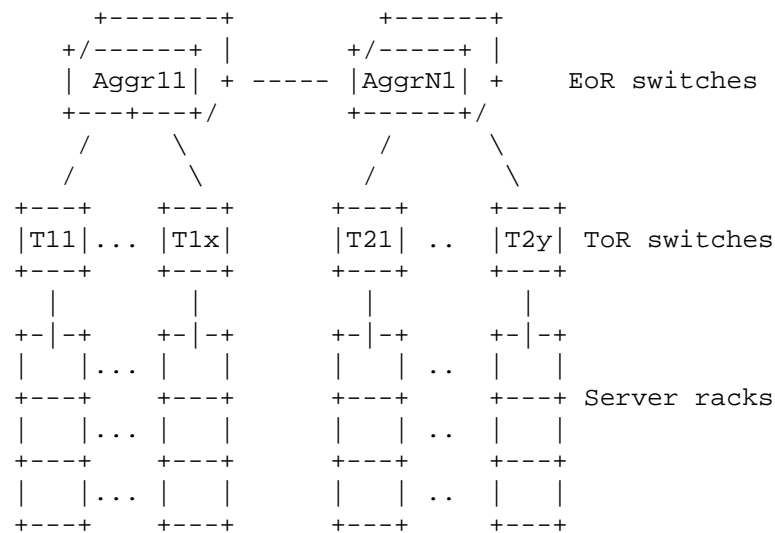
                Figure 1: Typical Data Center Network Design

   The following problems could occur when TRILL is deployed in a data
   center with a large number of end stations and when the end stations
   in one subnet/Label are placed under multiple edge RBridges:

      - Unnecessary filling of slots in the MAC address learning table
         of edge RBridges, e.g., RBridge T11, due to T11 receiving
         broadcast/multicast traffic (e.g., ARP/ND, cluster multicast,
         etc.) from end stations under other edge RBridges that are not
         actually communicating with any end stations attached to T11.

      - Packets being flooded across a TRILL campus when their
         destination MAC addresses are not in the ingress RBridge's MAC
         address to the egress RBridge cache.

3.2. Two Examples

   Consider a data center with 1,600 server racks. Each server rack has
   at least one ToR switch. The ToR switches are further divided into 8
   groups, with each group being connected by a set of aggregation
   switches. There could be 4 to 8 aggregation switches in each set to
   achieve load sharing for traffic to/from server racks. Let's
   consider the following two scenarios for the TRILL campus boundary if
   TRILL is deployed in this data center environment:

   - Scenario #1: TRILL campus boundary starts at the ToR switches:

      If each server rack has one ToR, there are 1,600 edge RBridges.
      If each rack has two ToR switches, then there will be 3,200
      edge RBridges.

      In this scenario, the TRILL campus will have more than 1,600
      (or 3,200) + 8*4 (or 8*8) nodes, which is a large IS-IS area.
      Even though a mesh IS-IS area can scale up to thousands of
      nodes, it is challenging for aggregation switches to handle
      IS-IS link state advertisement among hundreds of parallel
      ports.

      If each ToR has 40 downstream ports facing servers and each
      server has 10 VMs, there could be 40*10 = 400 end stations
      attached. If those end stations belong to 8 Labels, then the
      total number of MAC&Label entries learned by each edge RBridge
      in the worst case might be 400*8 = 3,200, which is not a large
      number.

   - Scenario #2: TRILL campus boundary starts at the aggregation
      switches:

      With the same assumptions as before, the number of nodes in the
      TRILL campus will be less than 100, and aggregation switches
      don't have to handle IS-IS link state advertisements among
      hundreds of parallel ports.

      However, the number of MAC&Label <-> Egress RBridge mapping
      entries to be learned and managed by the RBridge edge node can
      be very large. In the example above, each edge RBridge has 200
      edge ports facing the ToR switches. If each ToR has 40
      downstream ports facing servers and each server has 10 VMs,
      there could be 200*40*10 = 80,000 end stations attached. If
      all those end stations belong to 1,600 Labels (50 per Data
      Label) and each Data Label has 200 end stations, then under the

          worst-case scenario, the total number of MAC&Label entries to
          be learned by each edge RBridge can be 1,600*200=320,000, which
          is very large.

4. Benefits of Directory-Assisted TRILL Edge

   In some environments, particularly data centers, the assignment of
   applications to servers, including rack and row selection, is
   orchestrated by Server (or VM) Management System(s). That is, there
   is a database or multiple databases that have the knowledge of where
   each application is placed. If the application location information
   can be fed to RBridge edge nodes through some form of directory
   service, then there is much less chance of RBridge edge nodes
   receiving unknown MAC destination addresses, therefore less chance of
   flooding.

   Avoiding unknown unicast address flooding to the TRILL campus is
   especially valuable in the data center environment, because there is
   a higher chance of an edge RBridge receiving packets with an unknown
   unicast destination address and broadcast/multicast messages due to
   VM migration and servers being loaded with different applications.
   When a VM is moved to a new location or a server is loaded with a new
   application with a different IP/MAC addresses, it is more likely that
   the destination address of data packets sent out from those VMs is
   unknown to their attached edge RBridges. In addition, gratuitous ARP
   (IPv4 [RFC826]) or Unsolicited Neighbor Advertisement (IPv6
   [RFC4861]) sent out from those newly migrated or activated VMs have
   to be flooded to other edge RBridges that have VMs in the same
   subnets.

   The benefits of using directory assistance include:

      - Avoids flooding an unknown unicast destination address across
         the TRILL campus. The directory-enforced MAC&Label <-> Egress
         RBridge mapping table can determine if a data packet needs to
         be forwarded across the TRILL campus.

         When multiple RBridge edge ports are connected to end stations
         (servers/VMs), possibly via bridged LANs, a directory-assisted
         edge RBridge won't need to flood unknown unicast destination
         data frames to all ports of the edge RBridges in the frame's
         Data Label when it ingresses a frame. It can depend on the
         directory to locate the destination. When the directory
         doesn't have the needed information, the frames can be dropped
         or flooded depending on the policy configured.

- Reduces flooding of decapsulated Ethernet frames with an
  unknown MAC destination address to a bridged LAN connected to
  RBridge edge ports.

  When an RBridge receives a unicast TRILL data packet whose
  destination Nickname matches with its own, the normal procedure
  is for the RBridge to decapsulate it and forward the
  decapsulated Ethernet frame to the directly attached bridged
  LAN. If the destination MAC is unknown, the RBridge floods the
  decapsulated Ethernet frame out all ports in the frame's Data
  Label. With directory assistance, the egress RBridge can
  determine if the MAC destination address in a frame matches any
  end stations attached via the bridged LAN. Frames can be
  discarded if their destination addresses do not match.

- Reduces the amount of MAC&Label <-> Egress RBridge mapping
  maintained by edge RBridges. There is no need for an edge
  RBridge to keep MAC entries of remote end stations that don't
  communicate with the end stations locally attached.

- Eliminates ARP/ND being broadcast or multicast through the
  TRILL core.

- Provides some protection against spoofing of source addresses
  (see Section 7).

5. Generic Operation of Directory Assistance

   There are two different models for directory assistance to edge
   RBridges: Push Model and Pull Model. The directory information is
   described in Section 5.1 below, while Section 5.2 discusses Push
   Model requirements, and Section 5.3 Pull Model requirements.

5.1. Information in Directory for Edge RBridges

   To achieve the benefits of directory assistance for TRILL, the
   corresponding Directory Server entries will need, at a minimum, the
   following logical data structure:

   [IP, MAC, Data Label, {list of attached RBridge nicknames}, {list of
   interested RBridges}]

   The {list of attached RBridges} are the edge RBridges to which the
   host (or VM) is attached as specified by the [IP, MAC, Data Label] in
   the entry. The {list of interested RBridges} are the remote RBridges
   that might have attached hosts that communicate with the host in this
   entry.

When a host has multiple IP addresses, there will be multiple
entries.

The {list of interested RBridges} could get populated when an RBridge
queries for information, or information is pushed from a Directory
Server. The list is used to notify those RBridges when the host
(specified by the [IP, MAC, Data Label]) in the entry changes its
RBridge attachment. An explicit list in the directory is not needed
as long as the interested RBridges can be determined.

5.2. Push Model and Requirements

Under this model, Directory Server(s) push the MAC&Label <-> Egress
RBridge mapping for all the end stations that might communicate with
end stations attached to an RBridge edge node. If the packet's
destination address can't be found in the MAC&Label <-> Egress
RBridge table, the Ingress RBridge could be configured to:

     simply drop a data packet,

     flood it to the TRILL campus, or

     start the pull process to get information from the Pull Directory
     Server(s).

It may not be necessary for every edge RBridge to get the entire
mapping table for all the end stations in a campus. There are many
ways to narrow the full set down to a smaller set of remote end
stations that communicate with end stations attached to an edge
RBridge. A simple approach is to only push the mapping for the Data
Labels that have active end stations under an edge RBridge. This
approach can reduce the number of mapping entries being pushed.

However, the Push Model will usually push more entries of MAC&Label
<-> Egress RBridge mapping to an edge RBridges than needed. Under
the normal process of edge RBridge cache aging and unknown
destination address flooding, rarely used mapping entries would have
been removed. But it can be difficult for Directory Servers to
predict the communication patterns among applications within one Data
Label. Therefore, it is likely that the Directory Servers will push
down all the MAC&Label entries if there are end stations in the Data
Label attached to the edge RBridge. This is a disadvantage of the
Push Model compared with the Pull Model described below.

In the Push Model, it is necessary to have a way for an RBridge node
to request Directory Server(s) to push the mapping entries. This
method should at least include the Data Labels enabled on the
RBridge, so that the Directory Server doesn't need to push down the

entire set of mapping entries for all the end stations in the campus.
An RBridge must be able to get mapping entries when it is initialized
or restarted.

The Push Model's detailed method and any handshake mechanism between
an RBridge and Directory Server(s) is beyond the scope of this
framework document.

When a Directory Server needs to push a large number of entries to
edge RBridges, efficient data organization should be considered, for
example, with one edge RBridge nickname being associated with all the
attached end stations' MAC addresses and Data Labels. As shown in
Table 1 below, to make the data more compact, a representation can be
used where a nickname need only occur once for a set of Labels, each
of which occurs only once and each of which is associated with a set
of multiple IP and MAC address pairs. It would be much more bulky to
have each IP and MAC address pair separately accompanied by its Label
and by the nickname of the RBridge by which it is reachable.

```
     +------------+---------+------------------------------+
     | Nickname1  |Label-1  | IP/MAC1, IP/MAC2, ,, IP/MACn |
     |            |-------- +------------------------------+
     |            |Label-2  | IP/MAC1, IP/MAC2, ,, IP/MACn |
     |            |-------- +------------------------------+
     |            | ......  | IP/MAC1, IP/MAC2, ,, IP/MACn |
     +------------+-------- +------------------------------+
     | Nickname2  |Label-1  | IP/MAC1, IP/MAC2, ,, IP/MACn |
     |            |-------- +------------------------------+
     |            |Label-2  | IP/MAC1, IP/MAC2, ,,IP/MACn  |
     |            |-------- +------------------------------+
     |            |         | IP/MAC1, IP/MAC2, ,, IP/MACn |
     +------------+-------- +------------------------------+
     | -------    |-------- +------------------------------+
     |            |         | IP/MAC1, IP/MAC2, ,, IP/MACn |
     +------------+-------- +------------------------------+
```

            Table 1: Summarized Table Pushed Down from Directory

Whenever there is any change in MAC&Label <-> Egress RBridge mapping
that can be triggered by end stations being added, moved, or
decommissioned, an incremental update can be sent to the edge
RBridges that are impacted by the change. Therefore, something like
a sequence number has to be maintained by Directory Servers and
RBridges. Detailed mechanisms will be specified in a separate
document.

5.3. Pull Model and Requirements

   Under this model, an RBridge pulls the MAC&Label <-> Egress RBridge
   mapping entry from the Directory Server when its cache doesn't have
   the entry. There are a couple of possibilities for triggering the
   pulling process:

      - The RBridge edge node can send a pull request whenever it
         receives an unknown MAC destination, or

      - The RBridge edge node can intercept all ARP/ND requests and
         forward them or appropriate requests to the Directory Server(s)
         that has the information on where the target end stations are
         located.

   The Pull Directory response could indicate that the address being
   queried is unknown or that the requestor is administratively
   prohibited from getting an informative response.

   By using a Pull Directory, a frame with an unknown MAC destination
   address doesn't have to be flooded across the TRILL campus and the
   ARP/ND requests don't have to be broadcast or multicast across the
   TRILL campus.

   The ingress RBridge can cache the response pulled from the directory.
   The timer for such a cache should be short in an environment where
   VMs move frequently. The cache timer could be configured by the
   Management System or sent along with the Pulled reply by the
   Directory Server(s). It is important that the cached information be
   kept consistent with the actual placement of addresses in the campus;
   therefore, there needs to be some mechanism by which RBridges that
   have pulled information that has not expired can be informed when
   that information changes or becomes invalid for other reasons.

   One advantage of the Pull Model is that edge RBridges can age out
   MAC&Label entries if they haven't been used for a certain configured
   period of time or a period of time provided by the directory.
   Therefore, each edge RBridge will only keep the entries that are
   frequently used, so its mapping table size will be smaller. Edge
   RBridges would query the Directory Server(s) for unknown MAC
   destination addresses in data frames or ARP/ND and cache the
   response. When end stations attached to remote edge RBridges rarely
   communicate with the locally attached end stations, the corresponding
   MAC&VLAN entries would be aged out from the RBridge's cache.

   An RBridge waiting for a response from Directory Servers upon
   receiving a data frame with an unknown destination address is similar
   to an Layer-3/Layer-2 boundary router waiting for an ARP or ND

response upon receiving an IP data packet whose destination IP is not
in the router's IP/MAC cache table. Most deployed routers today do
hold the packet and send ARP/ND requests to the target upon receiving
a packet with a destination IP not in its IP-to-MAC cache. When
ARP/ND replies are received, the router will send the data packet to
the target. This practice minimizes flooding when targets don't
exist in the subnet.

When the target doesn't exist in the subnet, routers generally resend
an ARP/ND request a few more times before dropping the packets. So,
if the target doesn't exist in the subnet, the router's holding time
to wait for an ARP/ND response can be longer than the time taken by
the Pull Model to get IP-to-MAC mapping from a Directory Server.

RBridges with mapping entries being pushed from a Directory Server
can be configured to use the Pull Model for targets that don't exist
in the mapping data being pushed.

A separate document will specify the detailed messages and mechanism
for RBridges to pull information from Directory Server(s).

6. Recommendation

   TRILL should provide a directory-assisted approach. This document
   describes a basic framework for directory assistance to RBridge edge
   nodes. More detailed mechanisms will be described in a separate
   document or documents.

7. Security Considerations

   For general TRILL security considerations, see Section 6 of
   [RFC6325].

   Accurate mapping of IP addresses into MAC addresses and of MAC
   addresses to the RBridges from which they are reachable is important
   to the correct delivery of information. The security of specific
   directory-assisted mechanisms for delivering such information will be
   discussed in the document or documents specifying those mechanisms.

   A directory-assisted TRILL edge can be used to substantially improve
   the security of a TRILL campus over TRILL's default MAC address
   learning from the data plane. Assume S is an end station attached to
   RB1 trying to spoof a target end station T and that T is attached to
   RB2. Perhaps S wants to steal traffic intended for T or forge
   traffic as if it was from T.

With that default TRILL data-plane learning as described in
[RFC6325], S can impersonate T or any other end station in the same
Data Label (VLAN or FGL [FGL]) as S and possibly other Data Labels,
depending on how tightly VLAN admission and Appointed Forwarders
[RFC6439] are configured at the port by which S is connected to RB1.
S can just send native frames with the forged source MAC addresses of
T, perhaps broadcast frames for maximum effectiveness. With this
technique, S will frequently receive traffic intended for T and S can
easily forge traffic as being from T.

Such spoofing can be prevented to the extent that the network
RBridges (1) use trusted directory services as described above in
this document, (2) discard native frames received from a local end
station when the directory says that end stations should be remote,
and, (3) when appropriate, intercept ARP and ND messages and respond
locally. Under these circumstances, S would be limited to spoofing
targets on the same RBridge as the ingress RBridge for S (that is,
RB1 = RB2). RB1 would still need to learn which local end stations
were attached to which port, and S could confuse RB1 by sending
frames with the forged source MAC address of other end stations on
RB1. Although it would also still be restricted to frames in a VLAN
that would both be admitted by S's port of attachment and for which
that port is an Appointed Forwarder.

Security against spoofing could be even further strengthened by
adding port of attachment information to the directory and discarding
native frames that are received on the wrong port. This would limit
S to spoofing targets that were on the same link as S and in a VLAN
admitted by the port of that link's attachment to RB1 and for which
that port is an Appointed Forwarder (or, if the link is multiply
connected, in the same way at all of the ports by which the link is
attached to an RBridge).

Even without directory services, secure ND [RFC3971] or use of secure
ESADI (as described in [ESADI]) may also be helpful to security.

8. Acknowledgements

   Thanks for comments and review from the following:

   Sam Aldrin, David Black, Charlie Kaufman, Yizhou Li, and Erik
   Nordmark

9. Informative References

   [802.1Q]   IEEE Std 802.1Q-2011, "IEEE Standard for Local and
               metropolitan area networks - Virtual Bridged Local Area
               Networks", May 2011.

   [IS-IS]   ISO/IEC, "Intermediate System to Intermediate System
               intra-domain routeing information exchange protocol for
               use in conjunction with the protocol for providing the
               connectionless-mode network service (ISO 8473)", ISO/IEC
               10589:2002.

   [RFC826]   Plummer, D., "Ethernet Address Resolution Protocol: Or
               Converting Network Protocol Addresses to 48.bit Ethernet
               Address for Transmission on Ethernet Hardware", STD 37,
               RFC 826, November 1982.

   [RFC3971]   Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander,
               "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

   [RFC4861]   Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
               "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
               September 2007.

   [RFC6325]   Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A.
               Ghanwani, "Routing Bridges (RBridges): Base Protocol
               Specification", RFC 6325, July 2011.

   [RFC6326]   Eastlake, D., Banerjee, A., Dutt, D., Perlman, R., and A.
               Ghanwani, "Transparent Interconnection of Lots of Links
               (TRILL) Use of IS-IS", RFC 6326, July 2011.

   [RFC6439]   Perlman, R., Eastlake, D., Li, Y., Banerjee, A., and F.
               Hu, "Routing Bridges (RBridges): Appointed Forwarders",
               RFC 6439, November 2011.

   [ESADI]   Zhai, H., Hu, F., Perlman, R., Eastlake 3rd, D., and O.
               Stokes, "TRILL (Transparent Interconnection of Lots of
               Links): ESADI (End Station Address Distribution
               Information) Protocol", Work in Progress, July 2013.

   [FGL]   Eastlake 3rd, D., Zhang, M., Agarwal, P., Perlman, R., and
               D. Dutt, "TRILL (Transparent Interconnection of Lots of
               Links): Fine-Grained Labeling", Work in Progress, May
               2013.

Authors' Addresses

   Linda Dunbar
   Huawei Technologies
   5430 Legacy Drive, Suite #175
   Plano, TX 75024, USA

   Phone: +1-469-277-5840
   EMail: ldunbar@huawei.com

   Donald Eastlake
   Huawei Technologies
   155 Beaver Street
   Milford, MA 01757 USA

   Phone: +1-508-333-2270
   EMail: d3e3e3@gmail.com

   Radia Perlman
   Intel Labs
   2200 Mission College Blvd.
   Santa Clara, CA 95054-1549 USA

   Phone: +1-408-765-8080
   EMail: Radia@alum.mit.edu

   Igor Gashinsky
   Yahoo
   45 West 18th Street 6th floor
   New York, NY 10011 USA

   EMail: igor@yahoo-inc.com