Network Working Group
Internet Draft
draft-kompella-ppvpn-dtls-03.txt
Expiration Date: April 2004

K. Kompella (Juniper)
J. Achirica (Telefonica)
L. Andersson (Utfors)
V. Kompella (TiMetra)
K. Kumaki (KDDI)
M. Lasserre (Riverstone)
P. Lin (Yipes)
P. Menezes (Terabeam)
A. Moranganti (Appian)
H. Ould-Brahim (Nortel)
R. Raszuk (Cisco)
Y. Serbest (SBC)
S. Yeong-il (Korea Tel)

Decoupled Virtual Private LAN Services

Status of this Memo

Copyright Notice

Abstract

Virtual Private LAN Service (VPLS, also known as Transparent LAN Service) is a useful Service Provider offering. A common scenario where this offering is made is in the metro area, where several Multi-Tenant buildings are connected to an Office, and several Offices are connected by a metro area network. The Service Provider places a simple, low-cost box (MTU) in each building; these MTUs have uplinks to some box (PE) in one or more Offices. In such a scenario, it is useful to decouple the functions needed to offer VPLS across the MTU and PE; this document proposes one way of accomplishing that.

1. Introduction

Virtual Private LAN Service (VPLS, also known as Transparent LAN Service) is a useful Service Provider offering. A Virtual Private LAN appears in (almost) all respects as a LAN to the Service Provider customer; it is "transparent" in the sense of "transparent" bridging. However, in a VPLS, the customers are not all connected to a single LAN; the customers may be spread across a metro or wide area. In essence, a VPLS glues several individual LANs across a metro area to appear and function as a single LAN. See also [VPLS-REQ].

A common scenario where VPLS is offered is in the metro area, where several Multi-Tenant buildings are connected to an Office, and several Offices are connected by a metro area network, for example, a SONET ring. The Service Provider places a simple, low-cost box (which we will call an L2PE, also known as an MTU) in each building. A typical L2PE has 10-100 Ethernet ports that connect to customers (we call these the "customer-facing ports"), and one or more uplinks to a Service Provider aggregation box (PE) (we call these the "WAN" links, although they may either metro area or wide area links). The customer hand-off is thus an Ethernet; the L2PE acts as a metro/wide area bridge connecting the LAN at a customer site to other L2PEs that are connected to other LANs belonging to the same customer.

In such a scenario, it is useful to decouple the functions needed to offer VPLS across the L2PE and PE. These functions comprise learning MAC addresses, including learning across the metro area from other L2PEs; building a spanning tree, both on the LAN side and on the metro side; and discovering other L2PEs connected to a given customer. This document suggests that the former two functions (layer 2 functions) be run on the L2PE, and that the last function (layer 3) be run on the PE. In addition, there needs to be some information exchange between the PE and its L2PEs. This document does not suggest that decoupling is appropriate for all scenarios.

The spanning tree algorithm is useful in the case that an L2PE is connected to more than one PE which can reach a destination L2PE. This gives the L2PE a means to choosing one of the PEs as the primary means of reaching the destination L2PE, while preserving the others as backup paths. Alternative means of accomplishing this objective may be possible, and indeed, preferable, and are not ruled out by this document.

1.1. Motivation for Decoupling

We consider three ways of splitting the VPLS functions across the L2PE and PE: run all VPLS functions on the L2PE; run all VPLS functions on the PE; or run learning and spanning tree on the L2PE, and discovery on the PE, and some information exchange between them. In this subsection, we examine the repercussions of each.

1.1.1. Function Compatibility

In this document, we assume that an L2PE is a simple, low-cost layer 2 device, usually based on a bridge. That implies that an L2PE is designed to learn MAC addresses, and to run a spanning tree algorithm. An L2PE does not usually run complex layer 3 protocols, in particular, BGP, which is the de facto VPN discovery protocol.

On the other hand, a PE device is a more complex layer 3 device. Besides being connected to L2PEs and offering VPLS, a PE may offer several other services. For example, a PE may be directly connected to customers through WAN circuits to offer various VPN services, and/or public Internet access; and a PE may take part in peering sessions. This requires the PE to be capable of VPN discovery, and to run BGP, so it is natural for a PE to do VPLS discovery.

1.1.2. Scalability

Let's posit the following hypothetical numbers to get a feel for the scaling requirements. The numbers are just orders of magnitude; there is no claim that these are realistic, or supported by market data.

# of end hosts per VPLS	100-1000
# of TLS's per L2PE	10-100
# of L2PEs per PE	10-100
# of PEs in a metro area	10-100

Armed with these, let's examine the effect of running each of the VPLS functions at an L2PE vs. at the PE.

The number of MAC addresses that would need to be learned by an L2PE

ranges from 1000 to 100,000. The number that would need to be learned by a PE ranges from 10,000 to 10,000,000.

A spanning tree may need to be built for each VPLS. If so, the number of spanning trees that would need to be built by an L2PE ranges from 10 to a 100. The number of spanning trees that would need to be built by a PE ranges from 100 to 10,000.

The number of "discovery peers" (i.e., BGP peers if BGP is the discovery protocol) for an L2PE doing discovery ranges from 100 to 10,000. The number of peers for a PE doing discovery ranges from 10 to 100. (Note that the number of peers can be mitigated using techniques similar to those recommended for RFC 2547-style VPNs [IPVPN, IPVPNbis], e.g., partitioned route-reflector topologies.)

1.1.3. Configuration

Both the fact that the L2PE is a simple, low-cost box, and that there are one to two orders of magnitude more L2PEs than PEs suggest that the PE is more suitable as the place where VPLS should be configured. If all VPLS functions are run on the PE, this is automatically satisfied. If all VPLS functions are run on the L2PE, this is difficult to achieve. This document shows how this can be achieved in the case that the VPLS functions are distributed across the L2PE and PE.

2. Functional Requirements

This section presents the functional requirements of L2PEs and PEs in order to participate in a decoupled VPLS.

2.1. Requirements on the L2PE

An L2PE must be able to function as a transparent learning bridge, as it might well be connected to traditional bridges on customer-facing ports. Thus, it must be able to learn MAC addresses and run the spanning tree algorithm and flood packets when necessary. An L2PE must also be able to run the modified learning and spanning tree algorithms described below, as well as the modified flooding.

In addition, an L2PE must be able to send and receive labeled packets. This labeling could either be MPLS labels, or VLAN tagging. In the latter case, the VLAN tag could either be the first tag, or a stacked tag; unfortunately, there is no standard yet for VLAN stacking. It is absolutely necessary that the VLAN tag that is added is assigned by the Service Provider, and is independent of customer tags. More details on the use of VLAN tags between an L2PE and a PE

will be forthcoming in a future version of this document.

For MPLS labeling, an L2PE must be able to take an Ethernet frame from a customer-facing port, verify the CRC, strip the Ethernet preamble and CRC, and encapsulate the remaining Layer 2 frame as an MPLS packet with the appropriate label stack and send it to a PE. On the receiving side, the L2PE must be able to receive a labeled packet from a PE, and based on the label decide which VPLS the enclosed Ethernet frame belongs to; if necessary, learn the origin of the source MAC address (as described in the section on modified learning); remove the MPLS encapsulation and label; and generate a CRC and send the packet on the appropriate customer-facing port. Furthermore, the L2PE must be able to distinguish from which PE it received the labeled packet, i.e., it must support per-interface labels, for reasons described below.

Also, an L2PE should have some means to be provisioned and to monitor its operation. This requires some form of access to the L2PE; for example, if the provisioning and monitoring is to be done via SNMP, the L2PE would have to be IP-addressable, and to support SNMP functions.

Finally, an L2PE must run the L2PE-PE protocol described below, and to carry out the actions required by the protocol.

In order to accomplish the above, the L2PE may maintain in some form a mapping from (learned) MAC addresses to customer ports (for traditional learning), and a mapping from MAC addresses to received labels (for the modified learning); these mappings constitute the L2PE's MAC address cache. Ideally, the L2PE should maintain a separate MAC address cache for each VPLS it is part of, although in principle MAC addresses being globally unique, this should not be necessary. Also, the L2PE must have a (configured) mapping from a customer port to a VPLS, and for convenience, a mapping from a VPLS to a list of customer ports.

2.2. Requirements on the PE

A PE must have the Layer 2 VPN functionality described in [L2VPN], with the modifications described below. This requires that a PE be able to run BGP; MPLS: i.e., send and receive labeled packets, and swap, push and pop labels; and that a PE have some form of tunnels to every other PE taking part in any VPLS in which this PE is participating.

A PE must also support the configuration of VPLSs, and be able to run the L2PE-PE protocol described below.

3. PE Operation

This section describes how a PE running a decoupled VPLS operates, that is, how the VPLS is configured, how members are discovered, how the PE forwards packets in a VPLS.

The discovery mechanism used here is a variant of the Layer 2 VPN discovery [L2VPN]. Familiarity with that document is a prerequisite to understanding the operation of decoupled VPLSs as described here. A common mechanism, independent of whether the VPLS is decoupled or not, is proposed in [VPLS-BGP]; the two approaches are well aligned. A mechanism for signaling VPLS using LDP is given in [VPLS-LDP].

3.1. Configuration

We first list the configuration information that L2PEs and PEs need, then we suggest a means of configuring this information.

As we said earlier, an L2PE has several Ethernet ports that connect to customers, and some uplinks to PEs. The L2PE needs to be told which of its customer-facing ports belong in which VPLS. More precisely, the L2PE needs to be told which VPLSs it is a member of, and which PEs it is connected to. For each <PE, VPLS> pair, the L2PE must be told what its "site ID" is for that <PE, VPLS>. Finally, for each VPLS, the L2PE must be given the list of its <physical port, VLAN>s that belong to the VPLS.

A PE needs to be told which L2PEs it is connected to (and over which link), and which VPLSs each such L2PE participates in.

The minimum information that a PE must be configured with is the following:

```
<<L2PE ID (router ID)>
 <connecting interface>
 <<VPLS ID> <L2PE site ID>
 <<VPLS ID> <L2PE site ID>
 ...
>
```

That is, the PE is given the L2PE ID (as an IP address), the interface over which it is connected to the L2PE, and a list of VPLSs that that L2PE participates in. For each VPLS in this list, the PE is given the VPLS ID, the L2PE's site ID in this VPLS. A VPLS ID is an 8 octet Route Distinguisher; a site ID is a two octet integer (see [L2VPN] for details).

If it is desired that all VPLS configuration be done at the PE, here

is how this can be accomplished. The PE is configured with the following information for each L2PE to which it is connected:

```
<<L2PE ID (router ID)>
 <connecting interface>
 <<VPLS ID> <L2PE site ID>
  <<L2PE port ID, VLAN tag> <L2PE port ID, VLAN tag> ...>>
 <<VPLS ID> <L2PE site ID>
  <<L2PE port ID, VLAN tag> <L2PE port ID, VLAN tag> ...>>
 ...
>
```

That is, in addition to the above, the PE is given, for each L2PE and each VPLS that the L2PE participates in, a list of customer facing port IDs and corresponding VLAN tags that belong to that VPLS. An L2PE port ID is a two octet integer; a VLAN tag is a two octet (12 bit) 802.1q tag.

The PE then transfers all information relevant to an L2PE to that L2PE using the L2PE-PE protocol described later. The PE transfers all information relevant to other PEs running VPLS to them using a mechanism similar to [L2VPN], described below.

3.2. Generated Information

A PE given the above configuration generates label ranges as in [L2VPN] consisting of a label base, an offset and a size. It is expected that for a VPLS, a single, contiguous label range with offset 0 would suffice; if not, a number of non-overlapping label ranges can be used, as described in [L2VPN]. Call these label ranges the WAN label ranges for the <L2PE, VPLS> pair.

For the same <L2PE, VPLS> pair, the PE must generate a second set of label ranges called the L2PE label ranges. Each L2PE label range for an <L2PE, VPLS> pair corresponds exactly to one of the WAN label ranges for the <L2PE, VPLS>; in fact, the only difference is the label base.

All WAN label ranges generated by a given PE must be non-overlapping; similarly, all L2PE label ranges generated by a given PE must be non-overlapping. It is possible that some WAN label range and some L2PE label range overlap; this depends on factors outside the scope of this document (such as whether per-interface or per-box labels are being used). An implementation may of course choose to ensure that PE WAN labels and L2PE labels are always non-overlapping.

3.2.1. Multipoint Uplinks

If the PE-L2PE link is a multipoint link (such as Ethernet), and there are multiple PEs on this link, there is the possibility that the L2PE label ranges from the PEs overlap. In this case, some mechanism is needed to resolve the overlap.

One solution is that PEs that are connected to L2PEs over a multipoint link multicast their messages to the L2PEs, say to the ALL-ROUTERS multicast address. If PE y hears that PE x is using a certain label range, it remembers that, and ensures that future label ranges that it sends to an L2PE don't overlap. If two PEs send overlapping label ranges simultaneously, the PE with the lower router ID wins the contention, and the other PE withdraws its label range and issues a non-overlapping range. Details will be forthcoming.

Other solutions may also be possible.

3.3. VPLS Discovery

The mechanism used for VPLS member discovery is that in [L2VPN], with the following changes: there is a new code-point for encapsulations for Decoupled VPLS; and the "routes" that are installed are different (described next). Furthermore, it is anticipated that there is no need for topology information, as VPLSs are fully meshed. For simplicity, it is assumed that a single label range with offset 0 is used for both WAN and L2PE label ranges for a given <L2PE, VPLS> pair.

Say L2PE U is attached to PE X, and that L2PE U participates in VPLS T. Suppose L2PE U' attached to PE X' also participates in VPLS T. Suppose further that the site ID for L2PE U in VPLS T is p, and the site ID for L2PE U' is p'. Suppose finally that there is a tunnel W from PE X to PE X' and a reverse tunnel W' from PE X' to PE X.

PE X announces a WAN label range to all other PEs with label base K, offset 0, and length k (k >= p, p') for <L2PE U, VPLS T>. According to [L2VPN], this tells PE X' to use label (K+p') to send packets from L2PE U' to L2PE U. PE X' announces a WAN label range to all other PEs with label base K', offset 0, and length k' (k' >= p, p') for <L2PE U', VPLS T>. This tells PE X to use label (K'+p) to send packets from L2PE U to L2PE U'.

PE X also sends an L2PE label range to L2PE U with label base A, offset 0, and length k for VPLS T. This tells L2PE U to use label (A+p') to send to the L2PE with site ID p' (i.e., L2PE U'); and also that packets received with label (A+p') belong to VPLS T, and were originated at the L2PE with site ID p'. Similarly, PE X' sends an

L2PE label range to L2PE U' with label base A', offset 0, and length k' for VPLS T.

PE X installs "L2PE routes" and "WAN routes". We describe the L2PE route for packets from L2PE U to L2PE U': for labeled packets arriving from L2PE U, swap incoming label (A+p') with (K'+p), then put the packet in tunnel W to PE X'. The WAN route is for packets from L2PE U' to L2PE U, and is as follows: for packets arriving on tunnel W' from PE X', remove the packets from the tunnel (if needed), then swap incoming label (K+p') with label (A+p').

To complete the path from L2PE U to L2PE U', the WAN route installed by PE X' is described. For packets arriving on tunnel W from PE X, remove the packets from the tunnel (if needed), then swap incoming label (K'+p) with (A'+p).

Figure 1 below depicts this scenario: L2PE U connected to PE X and L2PE U' connected to PE X'. It further depicts L2PE U dual homed to PE Z, and another L2PE V dual homed to the same PE, PE Y. For each PE, it shows the WAN label range announced to other PEs, as well as the L2PE label range sent to its L2PE. Note that a dual homed L2PE MUST be given distinct site IDs for each uplink, whether to the same PE or different PEs; and that a PE must allocate and announce WAN and L2PE label ranges for each L2PE that it is connected, even if it is the same L2PE over different links (such as PE Y and L2PE V).

Figure 1

```
       {A,0,k} <- +-------+       W  ->      +-------+ -> {A',0,k'}
      ============| PE X  |<****************>| PE X' |=======[L2PE U']
      |           +-------+       W' <-      +-------+
      |  announces {K,0,k} \                /  announces {K',0,k'}
      | for L2PE U, site ID p\              /   for L2PE U', site ID p'
      |                      ..............
   [L2PE U]                  .    core    .
      |                      ..............   announces {L,0,l} for
      | announces {M,0,m}   /              \ L2PE V, site ID q
      | L2PE U, site ID r   /      announces \        -> {B,0,l}
      |           +-------+       {L',0,l'}  +-------+=======[      ]
      ============| PE Z  |       for L2PE V, | PE Y  |       [L2PE V]
       {C,0,m} <- +-------+       site ID q' +-------+=======[      ]
                                                      -> {B',0,l'}
```

3.4. Packet Forwarding

All VPLS packets from/to the L2PE to/from its PE are sent as labeled packets: the entire Ethernet frame (minus preamble and checksum) is encapsulated in an MPLS frame with a single label. This label encodes both the VPLS to which the packet belongs, as well as the source (for packets received) or destination (for packets sent).

VPLS packets from PE to PE are sent as a single label MPLS packet in a PE-PE tunnel. Thus, the packet may be an MPLS packet with a two or more label stack; or it may be MPLS in a GRE or IPSec tunnel, etc.

Say an endpoint "behind" L2PE U decides to send a VPLS T packet to some endpoint "behind" L2PE U'. All L2PE U knows about L2PE U' is that its site ID is p', i.e., to send packets to L2PE U', L2PE U must use label (A+p') (how it arrives at this is described in the section on learning). The L2PE route at PE X states that the label (A+p') is swapped with (K'+p), and the packet is put in tunnel W to PE X'. When PE X' gets the packet, it removes the packet from the tunnel, and sees label (K'+p). PE X' swaps the label with (A'+p) per its L2PE route, and sends this packet to L2PE U'. Seeing label (A'+p), L2PE U' knows that the packet is a VPLS T packet which originated at L2PE U (i.e., the L2PE with site ID p).

4. L2PE Operation

This section describes the information that an L2PE gets from its PE, and how learning and spanning tree computations are done in a VPLS.

Modified flooding and learning in the case that the L2PE-PE labeling is done using VLANs is entirely analogous, and will be detailed in a future version of this document.

4.1. L2PE-PE Information Exchange

The PE sends the following information to attached L2PEs:

```
<<PE ID (router ID)> <L2PE ID (router ID)>
 <<VPLS ID> <L2PE site ID>
  <<L2PE label range> <L2PE label range> ...>
  <<L2PE port ID, VLAN tag> <L2PE port ID, VLAN tag> ...>>
 <<VPLS ID> <L2PE site ID>
  <<L2PE label range> <L2PE label range> ...>
  <<L2PE port ID, VLAN tag> <L2PE port ID, VLAN tag> ...>>
 ...
>
```

That is, the PE sends its own ID, the L2PE's ID (for sanity checking), and a list of VPLS IDs. For each VPLS, the PE tells the L2PE its site ID in the VPLS, a list of L2PE label ranges, and a list of customer ports in the VPLS. (The last list is for the L2PE's private consumption, and is optional.) An L2PE label range consists of a label base, a site ID offset, the number of sites in the range, and a bit vector of currently active sites. The length of the bit vector in octets is (number of sites in the range)/8.

The exact format of this information, and the protocol that is used to transfer this information, as well as a list of status codes sent by the L2PE in response will be specified in greater detail in a later version.

4.2. Modified Flooding

Flooding is used for example when an L2PE wants to send a packet, but doesn't have the packet destination MAC address in its MAC address cache; or when the destination MAC address is a broadcast or multicast address. Such a packet may be received either from a customer port, or from a PE. The L2PE must first identify to which VPLS the packet belongs. It must then flood this packet, i.e., send a copy of the packet out on multiple ports. If the packet is received on a customer port, the L2PE must send a copy out on every other customer port in the VPLS, as well as to every other L2PE participating in the VPLS. If the packet is received from a PE, then the packet must be sent on every customer port; if the VPLS is a full mesh, the packet need not be sent to any other PE/L2PE. If the VPLS is not a full mesh, the L2PE must be told to which other PE/L2PEs it needs to flood the packet.

If an L2PE wants to flood a VPLS packet to all other L2PEs in the VPLS, the L2PE sends a copy of the packet with each label in the L2PE label ranges for that VPLS (except of course the label corresponding to the L2PE itself) and sends it to the PE. If an L2PE has multiple connections to PEs (the same or different), it MUST flood to each PE over each available connection.

4.3. Modified Learning

When L2PE x receives a packet from one of its PEs, it matches the incoming label with all the L2PE label ranges received from the sending PE. (Note: L2PE label ranges sent by a PE to an L2PE MUST be non-overlapping. However, an L2PE label range sent by one PE MAY overlap with an L2PE label range sent by another PE. An L2PE MUST be able to identify the interface over which a packet arrived, and use that information to disambiguate incoming labels.)

The incoming label tells L2PE x which VPLS this packet belongs to, and which L2PE it originated at (say x'). If the source MAC address S of the packet is not in the MAC address cache of L2PE x, L2PE x "learns" S by remembering the site ID of L2PE x' (or more simply, by associating S with the incoming label L). If at some future point, L2PE x wants to send a packet to destination MAC address S, it looks up its cache, sees that S maps to label L, pushes label L and sends it to its PE.

4.4. Modified Spanning Tree

An L2PE x will receive multiple copies of a flooded packet with a given source MAC address S from L2PE y if either L2PE x or L2PE y (or both) is multiply connected to PEs; each copy of the packet will be received with a distinct <incoming interface, L2PE label> pair. In that case, L2PE x must choose which <outgoing interface, L2PE label> pair it will use to get to MAC address S. This situation corresponds exactly to there being multiple ports connecting two bridges. If the VPLS is fully meshed, L2PE x can choose on its own which <interface, label> to use to send packets to MAC address S; i.e., L2PEs need not run a spanning tree algorithm across the metro or wide area which may be a distinct benefit.

If a VPLS is not fully meshed, each L2PE in the VPLS may treat multiple incoming L2PE labels for a given MAC address as multiple parallel ports to the same "bridge" (i.e., remote L2PE) and use the spanning tree algorithm to pick the active port (i.e., label) to use to reach the MAC address.

Finally, L2PE may use an IGP (or configuration) to decide which of the multiple incoming L2PE labels to use to map a given source MAC address.

4.5. Load Balancing

If an L2PE is multi-homed (i.e., it has multiple uplinks to one or more PEs), it may decide to load balance across those links. This decision can be made locally by the L2PE. The main consideration is to make sure that packets to the same destination use the same path to avoid packet re-ordering. This can be tricky, as broadcast or multicast packets may go to the same destination. A reasonable approach is to use the same uplink for all packets in a given VPLS, but to use different uplinks for different VPLSs, thus achieving some degree of load balancing.

5. Security Considerations

The security aspects of this solution will be discussed at a later time.

6. IANA Considerations

(To be filled in in a later revision.)

7. Acknowledgments

Many thanks to Brian Brown, Ian Hood, Vasile Radaoca and the Juniper "ethernet geeks", from whose stimulating discussions this idea was born; and to Yakov Rekhter whose probing questions helped clarify and refine the details.

8. References

[IPVPN] Rosen, E., and Rekhter, Y., "BGP/MPLS VPNs", RFC 2547, March 1999.

[IPVPNbis] Rosen, E., et al, "BGP/MPLS VPNs", work in progress.

[L2VPN] Kompella, K., et al, "Layer 2 VPNs Over Tunnels", work in progress.

[VPLS-BGP] Kompella, K., et al, "Virtual Private LAN Service", work in progress.

[VPLS-LDP] Kompella, V., et al, "Virtual Private Switched Network Services over an MPLS Network", work in progress.

[VPLS-REQ] Augustyn et al, "Requirements for Virtual Private LAN Services (VPLS)", work in progress.

9. IPR Notice

10. Full Copyright Statement

11. Author Information

Yeong-il,Seo
Korea Telecom
Telecommunication Network Laboratory
Member of Technical Staff
463-1 Junmin-dong, Yusung-gu, Taejeon, Korea
Tel : +82-42-870-8333 / FAX : +82-42-870-8339
Mobile : 016-235-0135 / E-mail : syi@hana.ne.kr

Yetik Serbest
SBC Communications
9505 Arboretum Blvd.
Austin TX 78759
serbest@tri.sbc.com

Robert Raszuk
cisco Systems
170 West Tasman Drive
San Jose, CA 94134
raszuk@cisco.com

Hamid Ould-Brahim
Nortel Networks
P O Box 3511 Station C
Ottawa ON K1Y 4H7 Canada
Phone: +1 (613) 765 3418
Email: hbrahim@nortelnetworks.com

Ashwin Moranganti
Appian Communications
email: amoranganti@appiancom.com
phone: 978 206-7248

Pascal Menezes
Terabeam Networks, Inc.
14833 NE 87th St.
Redmond, WA, USA
phone: (206) 686-2001
email: Pascal.Menezes@Terabeam.com

Pierre Lin
Yipes Communications, Inc.
114 Sansome St.

San Francisco CA 94104
email: pierre.lin@yipes.com

Marc Lasserre
Riverstone Networks
5200 Great America Parkway
Santa Clara CA 95054
marc@riverstonenet.com

Kenji Kumaki
IP Network Engineering Department
KDDI Corporation
KDDI Bldg. 2-3-2, Nishishinjuku, Shinjuku-ku, Tokyo
163-8003 Japan
+81-3-3347-6860 (phone)

Vach Kompella
TiMetra Networks
274 Ferguson Dr.
Mountain View, CA 94043
Email: vkompella@timetra.com

Loa Andersson
Utfors AB
Box 525, 169 29 Solna
Sweden
phone: +46 8 5270 5038
email: loa.andersson@utfors.se

Javier Achirica
Telefonica Data
javier.achirica@telefonica-data.com

Kireeti Kompella
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089
kireeti@juniper.net