Network Working Group
Request for Comments: 5565
Category: Standards Track

J. Wu
Y. Cui
Tsinghua University
C. Metz
E. Rosen
Cisco Systems, Inc.

June 2009

Softwire Mesh Framework

Status of This Memo

Copyright Notice

Abstract

The Internet needs to be able to handle both IPv4 and IPv6 packets. However, it is expected that some constituent networks of the Internet will be "single-protocol" networks. One kind of single-protocol network can parse only IPv4 packets and can process only IPv4 routing information; another kind can parse only IPv6 packets and can process only IPv6 routing information. It is nevertheless required that either kind of single-protocol network be able to provide transit service for the "other" protocol. This is done by passing the "other kind" of routing information from one edge of the single-protocol network to the other, and by tunneling the "other kind" of data packet from one edge to the other. The tunnels are known as "softwires". This framework document explains how the routing information and the data packets of one protocol are passed through a single-protocol network of the other protocol. The document is careful to specify when this can be done with existing technology and when it requires the development of new or modified technology.

Table of Contents

1. Introduction 3
2. Specification of Requirements 6
3. Scenarios of Interest 7
   3.1. IPv6-over-IPv4 Scenario 7
   3.2. IPv4-over-IPv6 Scenario 9
4. General Principles of the Solution 10
   4.1. E-IP and I-IP 10
   4.2. Routing 10
   4.3. Tunneled Forwarding 11
5. Distribution of Inter-AFBR Routing Information 11
6. Softwire Signaling 13
7. Choosing to Forward through a Softwire 15
8. Selecting a Tunneling Technology 15
9. Selecting the Softwire for a Given Packet 16
10. Softwire OAM and MIBs 17
   10.1. Operations and Maintenance (OAM) 17
   10.2. MIBs 18
11. Softwire Multicast 18
   11.1. One-to-One Mappings 18
        11.1.1. Using PIM in the Core 19
        11.1.2. Using mLDP and Multicast MPLS in the Core 20
   11.2. MVPN-Like Schemes 21
12. Inter-AS Considerations 22
13. Security Considerations 23
   13.1. Problem Analysis 23
   13.2. Non-Cryptographic Techniques 24

13.3. Cryptographic Techniques 26
14. References 27
14.1. Normative References 27
14.2. Informative References 28
15. Contributors 30
16. Acknowledgments 30

1. Introduction

The routing information in any IP backbone network can be thought of as being in one of two categories: "internal routing information" or "external routing information". The internal routing information consists of routes to the nodes that belong to the backbone, and to the interfaces of those nodes. External routing information consists of routes to destinations beyond the backbone, especially destinations to which the backbone is not directly attached. In general, BGP [RFC4271] is used to distribute external routing information, and an Interior Gateway Protocol (IGP) such as OSPF [RFC2328] or IS-IS [RFC1195] is used to distribute internal routing information.

Often an IP backbone will provide transit routing services for packets that originate outside the backbone and whose destinations are outside the backbone. These packets enter the backbone at one of its "edge routers". They are routed through the backbone to another edge router, after which they leave the backbone and continue on their way. The edge nodes of the backbone are often known as "Provider Edge" (PE) routers. The term "ingress" (or "ingress PE") refers to the router at which a packet enters the backbone, and the term "egress" (or "egress PE") refers to the router at which it leaves the backbone. Interior nodes are often known as "P routers". Routers that are outside the backbone but directly attached to it are known as "Customer Edge" (CE) routers. (This terminology is taken from [RFC4364].)

When a packet's destination is outside the backbone, the routing information that is needed within the backbone in order to route the packet to the proper egress is, by definition, external routing information.

Traditionally, the external routing information has been distributed by BGP to all the routers in the backbone, not just to the edge routers (i.e., not just to the ingress and egress points). Each of the interior nodes has been expected to look up the packet's destination address and route it towards the egress point. This is known as "native forwarding": the interior nodes look into each packet's header in order to match the information in the header with the external routing information.

It is, however, possible to provide transit services without requiring that all the backbone routers have the external routing information. The routing information that BGP distributes to each ingress router specifies the egress router for each route. The ingress router can therefore "tunnel" the packet directly to the egress router. "Tunneling the packet" means putting on some sort of encapsulation header that will force the interior routers to forward the packet to the egress router. The original packet is known as the "encapsulation payload". The P routers do not look at the packet header of the payload but only at the encapsulation header. Since the path to the egress router is part of the internal routing information of the backbone, the interior routers then do not need to know the external routing information. This is known as "tunneled forwarding". Of course, before the packet can leave the egress, it has to be decapsulated.

The scenario where the P routers do not have external routes is sometimes known as a "BGP-free core". That is something of a misnomer, though, since the crucial aspect of this scenario is not that the interior nodes don't run BGP, but that they don't maintain the external routing information.

In recent years, we have seen this scenario deployed to support VPN services, as specified in [RFC4364]. An edge router maintains multiple independent routing/addressing spaces, one for each VPN to which it interfaces. However, the routing information for the VPNs is not maintained by the interior routers. In most of these scenarios, MPLS is used as the encapsulation mechanism for getting the packets from ingress to egress. There are some deployments in which an IP-based encapsulation, such as L2TPv3 (Layer 2 Transport Protocol) [RFC3931] or GRE (Generic Routing Encapsulation) [RFC2784] is used.

This same technique can also be useful when the external routing information consists not of VPN routes, but of "ordinary" Internet routes. It can be used any time it is desired to keep external routing information out of a backbone's interior nodes, or in fact any time it is desired for any reason to avoid the native forwarding of certain kinds of packets.

This framework focuses on two such scenarios.

1. In this scenario, the backbone's interior nodes support only IPv6. They do not maintain IPv4 routes at all, and are not expected to parse IPv4 packet headers. Yet, it is desired to use such a backbone to provide transit services for IPv4 packets. Therefore, tunneled forwarding of IPv4 packets is

required. Of course, the edge nodes must have the IPv4 routes, but the ingress must perform an encapsulation in order to get an IPv4 packet forwarded to the egress.

2. This scenario is the reverse of scenario 1, i.e., the backbone's interior nodes support only IPv4, but it is desired to use the backbone for IPv6 transit.

In these scenarios, a backbone whose interior nodes support only one of the two address families is required to provide transit services for the other. The backbone's edge routers must, of course, support both address families. We use the term "Address Family Border Router" (AFBR) to refer to these PE routers. The tunnels that are used for forwarding are referred to as "softwires".

These two scenarios are known as the "Softwire Mesh Problem" [SW-PROB], and the framework specified in this document is therefore known as the "Softwire Mesh Framework". In this framework, only the AFBRs need to support both address families. The CE routers support only a single address family, and the P routers support only the other address family.

It is possible to address these scenarios via a large variety of tunneling technologies. This framework does not mandate the use of any particular tunneling technology. In any given deployment, the choice of tunneling technology is a matter of policy. The framework accommodates at least the use of MPLS ([RFC3031], [RFC3032]) -- both LDP-based (Label Distribution Protocol, [RFC5036]) and RSVP-TE-based (Resource Reservation Protocol - Traffic Engineering, [RFC3209]) -- L2TPv3 [RFC3931], GRE [RFC2784], and IP-in-IP [RFC2003]. The framework will also accommodate the use of IPsec tunneling, when that is necessary in order to meet security requirements.

It is expected that, in many deployments, the choice of tunneling technology will be made by a simple expression of policy, such as "always use IP-IP tunnels", or "always use LDP-based MPLS", or "always use L2TPv3".

However, other deployments may have a mixture of routers, some of which support, say, both GRE and L2TPv3, but others of which support only one of those techniques. It is desirable therefore to allow the network administration to create a small set of classes, and to configure each AFBR to be a member of one or more of these classes. Then the routers can advertise their class memberships to each other, and the encapsulation policies can be expressed as, e.g., "use L2TPv3 to tunnel to routers in class X; use GRE to tunnel to routers in class Y". To support such policies, it is necessary for the AFBRs to be able to advertise their class memberships; a standard way of doing this must be developed.

Policy may also require a certain class of traffic to receive a certain quality of service, and this may impact the choice of tunnel and/or tunneling technology used for packets in that class. This needs to be accommodated by the Softwire Mesh Framework.

The use of tunneled forwarding often requires that some sort of signaling protocol be used to set up and/or maintain the tunnels. Many of the tunneling technologies accommodated by this framework already have their own signaling protocols. However, some do not, and in some cases the standard signaling protocol for a particular tunneling technology may not be appropriate (for one or another reason) in the scenarios of interest. In such cases (and in such cases only), new signaling methodologies need to be defined and standardized.

In this framework, the softwires do not form an overlay topology that is visible to routing; routing adjacencies are not maintained over the softwires, and routing control packets are not sent through the softwires. Routing adjacencies among backbone nodes (including the edge nodes) are maintained via the native technology of the backbone.

There is already a standard routing method for distributing external routing information among AFBRs, namely BGP. However, in the scenarios of interest, we may be using IPv6-based BGP sessions to pass IPv4 routing information, and we may be using IPv4-based BGP sessions to pass IPv6 routing information. Furthermore, when IPv4 traffic is to be tunneled over an IPv6 backbone, it is necessary to encode the "BGP next hop" for an IPv4 route as an IPv6 address, and vice versa. The method for encoding an IPv4 address as the next hop for an IPv6 route is specified in [V6NLRI-V4NH]; the method for encoding an IPv6 address as the next hop for an IPv4 route is specified in [V4NLRI-V6NH].

2. Specification of Requirements

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 [RFC2119].

3. Scenarios of Interest

3.1. IPv6-over-IPv4 Scenario

In this scenario, the client networks run IPv6 but the backbone network runs IPv4. This is illustrated in Figure 1.

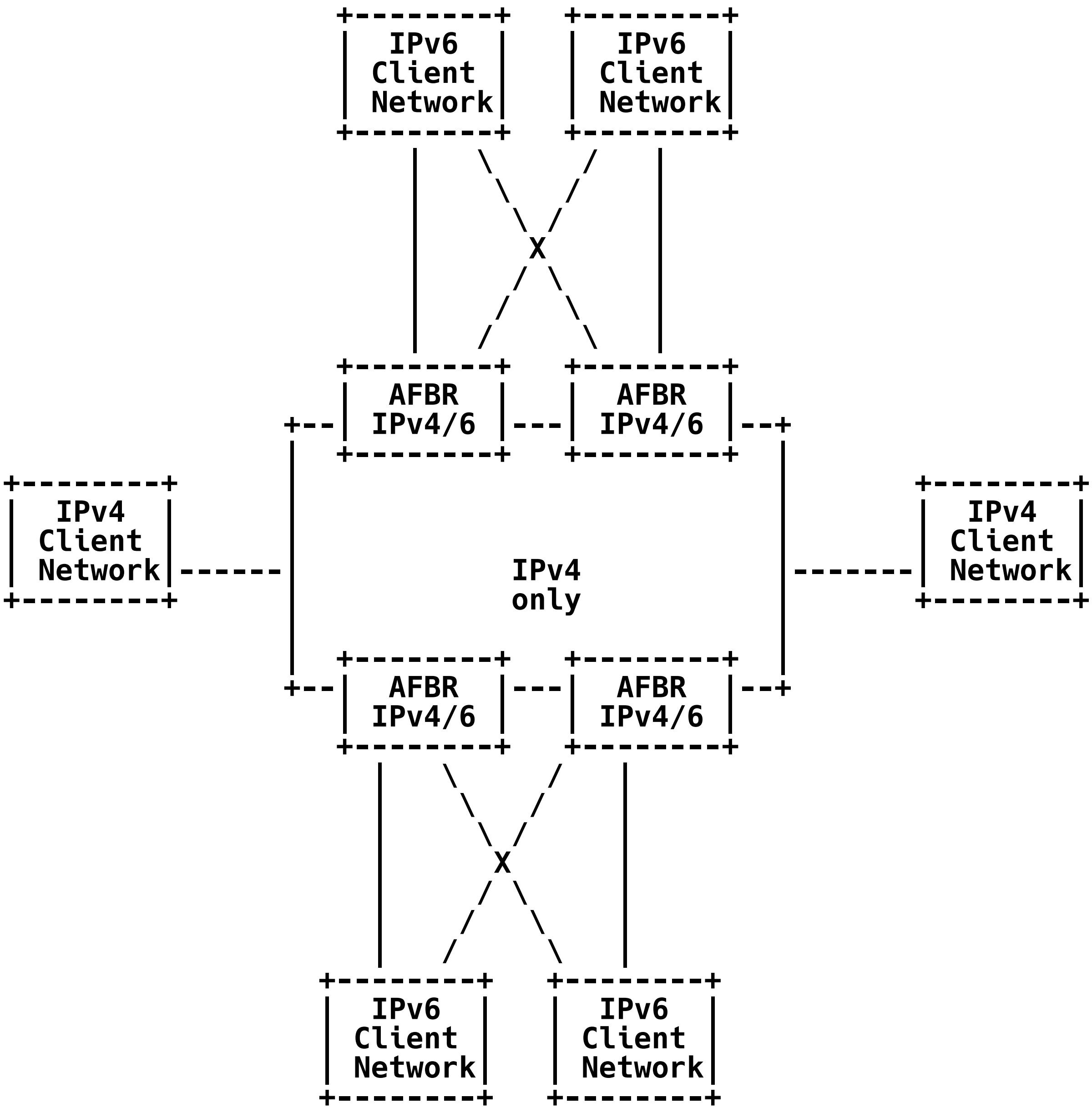

Figure 1: IPv6-over-IPv4 Scenario

The IPv4 transit core may or may not run MPLS. If it does, MPLS may be used as part of the solution.

While Figure 1 does not show any "backdoor" connections among the client networks, this framework assumes that there will be such connections. That is, there is no assumption that the only path between two client networks is via the pictured transit-core network. Hence, the routing solution must be robust in any kind of topology.

Many mechanisms for providing IPv6 connectivity across IPv4 networks have been devised over the past ten years. A number of different tunneling mechanisms have been used, some provisioned manually, and others based on special addressing. More recently, L3VPN (Layer 3 Virtual Private Network) techniques from [RFC4364] have been extended to provide IPv6 connectivity, using MPLS in the AFBRs and, optionally, in the backbone [V6NLRI-V4NH]. The solution described in this framework can be thought of as a superset of [V6NLRI-V4NH], with a more generalized scheme for choosing the tunneling (softwire) technology. In this framework, MPLS is allowed -- but not required -- even at the AFBRs. As in [V6NLRI-V4NH], there is no manual provisioning of tunnels, and no special addressing is required.

3.2. IPv4-over-IPv6 Scenario

In this scenario, the client networks run IPv4 but the backbone network runs IPv6. This is illustrated in Figure 2.

```
                          +--------+   +--------+
                          |  IPv4  |   |  IPv4  |
                          | Client |   | Client |
                          | Network|   | Network|
                          +--------+   +--------+
                              |   \     /   |
                              |    \   /    |
                              |     \ /     |
                              |      X      |
                              |     / \     |
                              |    /   \    |
                              |   /     \   |
                          +--------+   +--------+
                          |  AFBR  |   |  AFBR  |
                       +--| IPv4/6 |---| IPv4/6 |--+
                       |  +--------+   +--------+  |
       +--------+      |                           |       +--------+
       |  IPv6  |      |                           |       |  IPv6  |
       | Client |      |                           |       | Client |
       | Network|------|            IPv6           |-------| Network|
       +--------+      |            only           |       +--------+
                       |                           |
                       |  +--------+   +--------+  |
                       +--|  AFBR  |---|  AFBR  |--+
                          | IPv4/6 |   | IPv4/6 |
                          +--------+   +--------+
                            |   \     /   |
                            |    \   /    |
                            |     \ /     |
                            |      X      |
                            |     / \     |
                            |    /   \    |
                            |   /     \   |
                         +--------+   +--------+
                         |  IPv4  |   |  IPv4  |
                         | Client |   | Client |
                         | Network|   | Network|
                         +--------+   +--------+
```

Figure 2: IPv4-over-IPv6 Scenario

The IPv6 transit core may or may not run MPLS. If it does, MPLS may be used as part of the solution.

While Figure 2 does not show any "backdoor" connections among the client networks, this framework assumes that there will be such connections. That is, there is no assumption that the only path between two client networks is via the pictured transit-core network. Hence, the routing solution must be robust in any kind of topology.

While the issue of IPv6-over-IPv4 has received considerable attention in the past, the scenario of IPv4-over-IPv6 has not. Yet, it is a significant emerging requirement, as a number of service providers are building IPv6 backbone networks and do not wish to provide native IPv4 support in their core routers. These service providers have a large legacy of IPv4 networks and applications that need to operate across their IPv6 backbone. Solutions for this do not exist yet because it had always been assumed that the backbone networks of the foreseeable future would be dual stack.

4. General Principles of the Solution

This section gives a very brief overview of the procedures. The subsequent sections provide more detail.

4.1. E-IP and I-IP

In the following sections, we use the term "I-IP" (Internal IP) to refer to the form of IP (i.e., either IPv4 or IPv6) that is supported by the transit network. We use the term "E-IP" (External IP) to refer to the form of IP that is supported by the client networks. In the scenarios of interest, E-IP is IPv4 if and only if I-IP is IPv6, and E-IP is IPv6 if and only if I-IP is IPv4.

We assume that the P routers support only I-IP. That is, they are expected to have only I-IP routing information, and they are not expected to be able to parse E-IP headers. We similarly assume that the CE routers support only E-IP.

The AFBRs handle both I-IP and E-IP. However, only I-IP is used on AFBR's "core-facing interfaces", and E-IP is only used on its client-facing interfaces.

4.2. Routing

The P routers and the AFBRs of the transit network participate in an IGP for the purposes of distributing I-IP routing information.

The AFBRs use Internal BGP (IBGP) to exchange E-IP routing information with each other. Either there is a full mesh of IBGP connections among the AFBRs, or else some or all of the AFBRs are clients of a BGP Route Reflector. Although these IBGP connections are used to pass E-IP routing information (i.e., the Network Layer Reachability Information (NLRI) of the BGP updates is in the E-IP address family), the IBGP connections run over I-IP, and the BGP next hop for each E-IP NLRI is in the I-IP address family.

4.3. Tunneled Forwarding

When an ingress AFBR receives an E-IP packet from a client-facing interface, it looks up the packet's destination IP address. In the scenarios of interest, the best match for that address will be a BGP-distributed route whose next hop is the I-IP address of another AFBR, the egress AFBR.

The ingress AFBR must forward the packet through a tunnel (i.e, through a softwire) to the egress AFBR. This is done by encapsulating the packet, using an encapsulation header that the P routers can process and that will cause the P routers to send the packet to the egress AFBR. The egress AFBR then extracts the payload, i.e., the original E-IP packet, and forwards it further by looking up its IP destination address.

Several kinds of tunneling technologies are supported. Some of those technologies require explicit AFBR-to-AFBR signaling before the tunnel can be used, others do not.

Transmitting a packet through a softwire always requires that an encapsulation header be added to the original packet. The resulting packet is therefore always longer than the encapsulation payload. As an operational matter, the Maximum Transmission Unit (MTU) of the softwire's path SHOULD be large enough so that (a) no packet will need to be fragmented before being encapsulated, and (b) no encapsulated packet will need to be fragmented while it is being forwarded along a softwire. A general discussion of MTU issues in the context of tunneled forwarding may be found in [RFC4459].

5. Distribution of Inter-AFBR Routing Information

AFBRs peer with routers in the client networks to exchange routing information for the E-IP family.

AFBRs use BGP to distribute the E-IP routing information to each other. This can be done by an AFBR-AFBR mesh of IBGP sessions, but more likely is done through a BGP Route Reflector, i.e., where each AFBR has an IBGP session to one or two Route Reflectors rather than to other AFBRs.

The BGP sessions between the AFBRs, or between the AFBRs and the Route Reflector, will run on top of the I-IP address family. That is, if the transit core supports only IPv6, the IBGP sessions used to distribute IPv4 routing information from the client networks will run over IPv6; if the transit core supports only IPv4, the IBGP sessions used to distribute IPv6 routing information from the client networks will run over IPv4. The BGP sessions thus use the native networking layer of the core; BGP messages are NOT tunneled through softwires or through any other mechanism.

In BGP, a routing update associates an address prefix (or more generally, NLRI) with the address of a BGP next hop (NH). The NLRI is associated with a particular address family. The NH address is also associated with a particular address family, which may be the same as or different than the address family associated with the NLRI. Generally, the NH address belongs to the address family that is used to communicate with the BGP speaker to whom the NH address belongs.

Since routing updates that contain information about E-IP address prefixes are carried over BGP sessions that use I-IP transport, and since the BGP messages are not tunneled, a BGP update providing information about an E-IP address prefix will need to specify a next hop address in the I-IP family.

Due to a variety of historical circumstances, when the NLRI and the NH in a given BGP update are of different address families, it is not always obvious how the NH should be encoded. There is a different encoding procedure for each pair of address families.

In the case where the NLRI is in the IPv6 address family, and the NH is in the IPv4 address family, [V6NLRI-V4NH] explains how to encode the NH.

In the case where the NLRI is in the IPv4 address family, and the NH is in the IPv6 address family, [V4NLRI-V6NH] explains how to encode the NH.

If a BGP speaker sends an update for an NLRI in the E-IP family, and the update is being sent over a BGP session that is running on top of the I-IP network layer, and the BGP speaker is advertising itself as the NH for that NLRI, then the BGP speaker MUST, unless explicitly overridden by policy, specify the NH address in the I-IP family. The address family of the NH MUST NOT be changed by a Route Reflector.

In some cases (e.g., when [V4NLRI-V6NH] is used), one cannot follow this rule unless one's BGP peers have advertised a particular BGP capability. This leads to the following softwire deployment restriction: if a BGP capability is defined for the case in which an E-IP NLRI has an I-IP NH, all the AFBRs in a given transit core MUST advertise that capability.

If an AFBR has multiple IP addresses, the network administrators usually have considerable flexibility in choosing which one the AFBR uses to identify itself as the next hop in a BGP update. However, if the AFBR expects to receive packets through a softwire of a particular tunneling technology, and if the AFBR is known to that tunneling technology via a specific IP address, then that same IP address must be used to identify the AFBR in the next hop field of the BGP updates. For example, if L2TPv3 tunneling is used, then the IP address that the AFBR uses when engaging in L2TPv3 signaling must be the same as the IP address it uses to identify itself in the next hop field of a BGP update.

In [V6NLRI-V4NH], IPv6 routing information is distributed using the labeled IPv6 address family. This allows the egress AFBR to associate an MPLS label with each IPv6 address prefix. If an ingress AFBR forwards packets through a softwire that can carry MPLS packets, each data packet can carry the MPLS label corresponding to the IPv6 route that it matched. This may be useful at the egress AFBR, for demultiplexing and/or enhanced performance. It is also possible to do the same for the IPv4 address family, i.e., to use the labeled IPv4 address family instead of the IPv4 address family. The use of the labeled IP address families in this manner is OPTIONAL.

6. Softwire Signaling

A mesh of inter-AFBR softwires spanning the transit core must be in place before packets can flow between client networks. Given N dual-stack AFBRs, this requires N^2 "point-to-point IP" or "label switched path" (LSP) tunnels. While in theory these could be configured manually, that would result in a very undesirable O(N^2) provisioning problem. Therefore, manual configuration of point-to-point tunnels is not considered part of this framework.

Because the transit core is providing layer 3 transit services, point-to-point tunnels are not required by this framework; multipoint-to-point tunnels are all that is needed. In a multipoint-to-point tunnel, when a packet emerges from the tunnel there is no way to tell which router put the packet into the tunnel. This models the native IP forwarding paradigm, wherein the egress router cannot determine a given packet's ingress router. Of course, point-to-point tunnels might be required for some reason beyond the basic requirements described in this document. For example, Quality of

Service (QoS) or security considerations might require the use of point-to-point tunnels. So point-to-point tunnels are allowed, but not required, by this framework.

If it is desired to use a particular tunneling technology for the softwires, and if that technology has its own "native" signaling methodology, the presumption is that the native signaling will be used. This would certainly apply to MPLS-based softwires, where LDP or RSVP-TE would be used. An IPsec-based softwire would use standard IKEv2 (Internet Key Exchange) [RFC4306] and IPsec [RFC4301] signaling, as that is necessary in order to guarantee the softwire's security properties.

A GRE-based softwire might or might not require signaling, depending on whether various optional GRE header fields are to be used. GRE does not have any "native" signaling, so for those cases, a signaling procedure needs to be developed to support softwires.

Another possible softwire technology is L2TPv3. While L2TPv3 does have its own native signaling, that signaling sets up point-to-point tunnels. For the purpose of softwires, it is better to use L2TPv3 in a multipoint-to-point mode, and this requires a different kind of signaling.

The signaling to be used for GRE and L2TPv3 to cover these scenarios is BGP-based, and is described in [RFC5512].

If IP-IP tunneling is used, or if GRE tunneling is used without options, no signaling is required, as the only information needed by the ingress AFBR to create the encapsulation header is the IP address of the egress AFBR, and that is distributed by BGP.

When the encapsulation IP header is constructed, there may be fields in the IP whose value is determined neither by whatever signaling has been done nor by the distributed routing information. The values of these fields are determined by policy in the ingress AFBR. Examples of such fields may be the TTL (Time to Live) field, the DSCP (Diffserv Service Classes) bits, etc.

It is desirable for all necessary softwires to be fully set up before the arrival of any packets that need to go through the softwires. That is, the softwires should be "always on". From the perspective of any particular AFBR, the softwire endpoints are always BGP next hops of routes that the AFBR has installed. This suggests that any necessary softwire signaling should either be done as part of normal system startup (as would happen, e.g., with LDP-based MPLS) or else be triggered by the reception of BGP routing information (such as is described in [RFC5512]); it is also helpful if distribution of the routing information that serves as the trigger is prioritized.

7. Choosing to Forward through a Softwire

The decision to forward through a softwire, instead of to forward natively, is made by the ingress AFBR. This decision is a matter of policy.

In many cases, the policy will be very simple. Some useful policies are:

- If routing says that an E-IP packet has to be sent out a core-facing interface to an I-IP core, then send the packet through a softwire.

- If routing says that an E-IP packet has to be sent out an interface that only supports I-IP packets, then send the E-IP packet through a softwire.

- If routing says that the BGP next hop address for an E-IP packet is an I-IP address, then send the E-IP packet through a softwire.

- If the route that is the best match for a particular packet's destination address is a BGP-distributed route, then send the packet through a softwire (i.e., tunnel all BGP-routed packets).

More complicated policies are also possible, but a consideration of those policies is outside the scope of this document.

8. Selecting a Tunneling Technology

The choice of tunneling technology is a matter of policy configured at the ingress AFBR.

It is envisioned that, in most cases, the policy will be a very simple one, and will be the same at all the AFBRs of a given transit core -- e.g., "always use LDP-based MPLS" or "always use L2TPv3".

However, other deployments may have a mixture of routers, some of which support, say, both GRE and L2TPv3, but others of which support only one of those techniques. It is desirable therefore to allow the network administration to create a small set of classes and to configure each AFBR to be a member of one or more of these classes. Then the routers can advertise their class memberships to each other, and the encapsulation policies can be expressed as, e.g., "use L2TPv3 to talk to routers in class X; use GRE to talk to routers in class

Y". To support such policies, it is necessary for the AFBRs to be able to advertise their class memberships. [RFC5512] specifies a way in which an AFBR may advertise, to other AFBRS, various characteristics that may be relevant to the policy (e.g., "I belong to class Y"). In many cases, these characteristics can be represented by arbitrarily selected communities or extended communities, and the policies at the ingress can be expressed in terms of these classes (i.e., communities).

Policy may also require a certain class of traffic to receive a certain quality of service, and this may impact the choice of tunnel and/or tunneling technology used for packets in that class. This framework allows a variety of tunneling technologies to be used for instantiating softwires. The choice of tunneling technology is a matter of policy, as discussed in Section 1.

While in many cases the policy will be unconditional, e.g., "always use L2TPv3 for softwires", in other cases the policy may specify that the choice is conditional upon information about the softwire remote endpoint, e.g., "use L2TPv3 to talk to routers in class X; use GRE to talk to routers in class Y". It is desirable therefore to allow the network administration to create a small set of classes, and to configure each AFBR to be a member of one or more of these classes. If each such class is represented as a community or extended community, then [RFC5512] specifies a method that AFBRs can use to advertise their class memberships to each other.

This framework also allows for policies of arbitrary complexity, which may depend on characteristics or attributes of individual address prefixes as well as on QoS or security considerations. However, the specification of such policies is not within the scope of this document.

9. Selecting the Softwire for a Given Packet

Suppose it has been decided to send a given packet through a softwire. Routing provides the address, in the address family of the transport network, of the BGP next hop. The packet MUST be sent through a softwire whose remote endpoint address is the same as the BGP next hop address.

Sending a packet through a softwire is a matter of first encapsulating the packet with an encapsulation header that can be processed by the transit network and then transmitting towards the softwire's remote endpoint address.

In many cases, once one knows the remote endpoint address, one has all the information one needs in order to form the encapsulation header. This will be the case if the tunnel technology instantiating the softwire is, e.g., LDP-based MPLS, IP-in-IP, or GRE without optional header fields.

If the tunnel technology being used is L2TPv3 or GRE with optional header fields, additional information from the remote endpoint is needed in order to form the encapsulation header. The procedures for sending and receiving this information are described in [RFC5512].

If the tunnel technology being used is RSVP-TE-based MPLS or IPsec, the native signaling procedures of those technologies will need to be used.

If the packet being sent through the softwire matches a route in the labeled IPv4 or labeled IPv6 address families, it should be sent through the softwire as an MPLS packet with the corresponding label. Note that most of the tunneling technologies mentioned in this document are capable of carrying MPLS packets, so this does not presuppose support for MPLS in the core routers.

10. Softwire OAM and MIBs

10.1. Operations and Maintenance (OAM)

Softwires are essentially tunnels connecting routers. If they disappear or degrade in performance, then connectivity through those tunnels will be impacted. There are several techniques available to monitor the status of the tunnel endpoints (AFBRs) as well as the tunnels themselves. These techniques allow operations such as softwire path tracing, remote softwire endpoint pinging, and remote softwire endpoint liveness failure detection.

Examples of techniques applicable to softwire OAM include:

- BGP/TCP timeouts between AFBRs

- ICMP or LSP echo request and reply addressed to a particular AFBR

- BFD (Bidirectional Forwarding Detection) [BFD] packet exchange between AFBR routers

Another possibility for softwire OAM is to build something similar to [RFC4378] or, in other words, to create and generate softwire echo request/reply packets. The echo request sent to a well-known UDP port would contain the egress AFBR IP address and the softwire identifier as the payload (similar to the MPLS Forwarding Equivalence

Class contained in the LSP echo request). The softwire echo packet would be encapsulated with the encapsulation header and forwarded across the same path (inband) as that of the softwire itself.

This mechanism can also be automated to periodically verify remote softwire endpoint reachability, with the loss of reachability being signaled to the softwire application on the local AFBR, thus enabling suitable actions to be taken. Consideration must be given to the trade-offs between the scalability of such mechanisms versus the time required for detection of loss of endpoint reachability for such automated mechanisms.

In general, a framework for softwire OAM can, for a large part, be based on the [RFC4176] framework.

10.2. MIBs

Specific MIBs do exist to manage elements of the Softwire Mesh Framework. However, there will be a need to either extend these MIBs or create new ones that reflect the functional elements that can be SNMP-managed within the softwire network.

11. Softwire Multicast

A set of client networks, running E-IP, that are connected to a provider's I-IP transit core may wish to run IP multicast applications. Extending IP multicast connectivity across the transit core can be done in a number of ways, each with a different set of characteristics. Most (though not all) of the possibilities are either slight variations of the procedures defined for L3VPNs in [L3VPN-MCAST].

We will focus on supporting those multicast features and protocols that are typically used across inter-provider boundaries. Support is provided for PIM-SM (Protocol Independent Multicast - Sparse Mode) and PIM-SSM (PIM Source-Specific Mode). Support for BIDIR-PIM (Bidirectional PIM), BSR (Bootstrap Router Mechanism for PIM), and AutoRP (Automatic Rendezvous Point Determination) is not provided as these features are not typically used across inter-provider boundaries.

11.1. One-to-One Mappings

In the "one-to-one mapping" scheme, each client multicast tree is extended through the transit core so that for each client tree there is exactly one tree through the core.

The one-to-one scheme is not used in [L3VPN-MCAST] because it requires an amount of state in the core routers that is proportional to the number of client multicast trees passing through the core. In the VPN context, this is considered undesirable because the amount of state is unbounded and out of the control of the service provider. However, the one-to-one scheme models the typical "Internet multicast" scenario where the client network and the transit core are both IPv4 or both IPv6. If it scales satisfactorily for that case, it should also scale satisfactorily for the case where the client network and the transit core support different versions of IP.

11.1.1. Using PIM in the Core

When an AFBR receives an E-IP PIM control message from one of its CEs, it translates it from E-IP to I-IP, and forwards it towards the source of the tree. Since the routers in the transit core will not generally have a route to the source of the tree, the AFBR must include an "RPF (Reverse Path Forwarding) Vector" [RFC5496] in the PIM message.

Suppose an AFBR A receives an E-IP PIM Join/Prune message from a CE for either an (S,G) tree or a (*,G) tree. The AFBR would have to "translate" the PIM message into an I-IP PIM message. It would then send it to the neighbor that is the next hop along the route to the root of the (S,G) or (*,G) tree. In the case of an (S,G) tree, the root of the tree is S; in the case of a (*,G) tree, the root of the tree is the Rendezvous Point (RP) for the group G.

Note that the address of the root of the tree will be an E-IP address. Since the routers within the transit core (other than the AFBRs) do not have routes to E-IP addresses, A must put an RPF Vector [RFC5496] in the PIM Join/Prune message that it sends to its upstream neighbor. The RPF Vector will identify, as an I-IP address, the AFBR B that is the egress point in the transit network along the route to the root of the multicast tree. AFBR B is AFBR A's BGP next hop for the route to the root of the tree. The RPF Vector allows the core routers to forward PIM Join/Prune messages upstream towards the root of the tree, even though they do not maintain E-IP routes.

In order to translate an E-IP PIM message into an I-IP PIM message, the AFBR A must translate the address of S (in the case of an (S,G) group) or the address of G's RP from the E-IP address family to the I-IP address family, and the AFBR B must translate them back.

In the case where E-IP is IPv4 and I-IP is IPv6, it may be possible to do this translation algorithmically. A can translate the IPv4 S into the corresponding IPv4-mapped IPv6 address [RFC4291], and then B can translate it back. At the time of this writing, there is no such thing as an IPv4-mapped IPv6 multicast address, but if such a thing were to be standardized, then A could also translate the IPv4 G into IPv6, and B could translate it back. The precise circumstances under which these translations are to be done would be a matter of policy.

Obviously, this translation procedure does not generalize to the case where the client multicast is IPv6 but the core is IPv4. To handle that case, one needs additional signaling between the two AFBRs. Each downstream AFBR needs to signal the upstream AFBR that it needs a multicast tunnel for (S,G). The upstream AFBR must then assign a multicast address G' to the tunnel and inform the downstream of the P-G value to use. The downstream AFBR then uses PIM/IPv4 to join the (S',G') tree, where S' is the IPv4 address of the upstream ASBR (Autonomous System Border Router).

The (S',G') trees should be SSM trees.

This procedure can be used to support client multicasts of either IPv4 or IPv6 over a transit core of the opposite protocol. However, it only works when the client multicasts are SSM, since it provides no method for mapping a client "prune a source off the (*,G) tree" operation into an operation on the (S',G') tree. This method also requires additional signaling. The BGP-based signaling of [L3VPN-MCAST-BGP] is one signaling method that could be used. Other signaling methods could be defined as well.

11.1.2. Using mLDP and Multicast MPLS in the Core

LDP extensions for point-to-multipoint and multipoint-to-multipoint LSPs are specified in [MLDP]; we will use the term "mLDP" to refer to those LDP extensions. If the transit core implements mLDP and supports multicast MPLS, then client Source-Specific Multicast (SSM) trees can be mapped one-to-one onto P2MP (Point-to-Multipoint) LSPs.

When an AFBR A receives an E-IP PIM Join/Prune message for (S,G) from one of its CEs, where G is an SSM group, it would use mLDP to join a P2MP LSP. The root of the P2MP LSP would be the AFBR B that is A's BGP next hop on the route to S. In mLDP, a P2MP LSP is uniquely identified by a combination of its root and an "FEC (Forwarding Equivalence Class) identifier". The original (S,G) can be algorithmically encoded into the FEC identifier so that all AFBRs that need to join the P2MP LSP for (S,G) will generate the same FEC identifier. When the root of the P2MP LSP (AFBR B) receives such an mLDP message, it extracts the original (S,G) from the FEC identifier, creates an "ordinary" E-IP PIM Join/Prune message, and sends it to the CE that is its next hop on the route to S.

The method of encoding the (S,G) into the FEC identifier needs to be standardized. The encoding must be self-identifying so that a node that is the root of a P2MP LSP can determine whether a FEC identifier is the result of having encoded a PIM (S,G).

The appropriate state machinery must be standardized so that PIM events at the AFBRs result in the proper mLDP events. For example, if at some point an AFBR determines (via PIM procedures) that it no longer has any downstream receivers for (S,G), the AFBR should invoke the proper mLDP procedures to prune itself off the corresponding P2MP LSP.

Note that this method cannot be used when the G is a Sparse Mode group. The reason this method cannot be used is that mLDP does not have any function corresponding to the PIM "prune this source off the shared tree" function. So if a P2MP LSP were mapped one-to-one with a P2MP LSP, duplicate traffic could end up traversing the transit core (i.e., traffic from S might travel down both the shared tree and S's source tree). Alternatively, one could devise an AFBR-to-AFBR protocol to prune sources off the P2MP LSP at the root of the LSP. It is recommended, though, that client SM multicast groups be supported by other methods, such as those discussed below.

Client-side bidirectional multicast groups set up by PIM-bidir could be mapped using the above technique to MP2MP (Multipoint-to-Multipoint) LSPs set up by mLDP [MLDP]. We do not consider this further, as inter-provider bidirectional groups are not in use anywhere.

11.2. MVPN-Like Schemes

The "MVPN (Multicast VPN)-like schemes" are those described in [L3VPN-MCAST] and its companion documents (such as [L3VPN-MCAST-BGP]). To apply those schemes to the softwire environment, it is necessary only to treat all the AFBRs of a given transit core as if they were all, for multicast purposes, PE routers attached to the same VPN.

The MVPN-like schemes do not require a one-to-one mapping between client multicast trees and transit-core multicast trees. In the MVPN environment, it is a requirement that the number of trees in the core scales less than linearly with the number of client trees. This requirement may not hold in the softwire scenarios.

The MVPN-like schemes can support SM, SSM, and Bidir groups. They provide a number of options for the control plane:

- LAN-like

  Use a set of multicast trees in the core to emulate a LAN (Local Area Network) and run the client-side PIM protocol over that "LAN". The "LAN" can consist of a single Bidir tree containing all the AFBRs or a set of SSM trees, one rooted at each AFBR and containing all the other AFBRs as receivers.

- NBMA (Non-Broadcast Multiple Access), using BGP

  The client-side PIM signaling can be translated into BGP-based signaling, with a BGP Route Reflector mediating the signaling.

These two basic options admit of many variations; a comprehensive discussion is in [L3VPN-MCAST].

For the data plane, there are also a number of options:

- All multicast data sent over the emulated LAN. This particular option is not very attractive, though, for the softwire scenarios, as every AFBR would have to receive every client multicast packet.

- Every multicast group mapped to a tree that is considered appropriate for that group, in the sense of causing the traffic of that group to go to "too many" AFBRs that don't need to receive it.

Again, a comprehensive discussion of the issues can be found in [L3VPN-MCAST].

12. Inter-AS Considerations

We have so far only considered the case where a "transit core" consists of a single Autonomous System (AS). If the transit core consists of multiple ASes, then it may be necessary to use softwires whose endpoints are AFBRs attached to different Autonomous Systems. In this case, the AFBR at the remote endpoint of a softwire is not the BGP next hop for packets that need to be sent on the softwire. Since the procedures described above require the address of a remote softwire endpoint to be the same as the address of the BGP next hop, those procedures do not work as specified when the transit core consists of multiple ASes.

There are several ways to deal with this situation.

1. Don't do it; require that there be AFBRs at the edge of each AS so that a transit core does not extend more than one AS.

2. Use multi-hop EBGP to allow AFBRs to send BGP routes to each other, even if the ABFRs are not in the same or in neighboring ASes.

3. Ensure that an ASBR that is not an AFBR does not change the next hop field of the routes for which encapsulation is needed.

In the latter two cases, BGP recursive next hop resolution needs to be done, and encapsulations may need to be "stacked" (i.e., multiple layers of encapsulation may need to be used).

For instance, consider packet P with destination IP address D. Suppose it arrives at ingress AFBR A1 and that the route that is the best match for D has BGP next hop B1. So A1 will encapsulate the packet for delivery to B1. If B1 is not within A1's AS, A1 will need to look up the route to B1 and then find the BGP next hop, call it B2, of that route. If the interior routers of A1's AS do not have routes to B1, then A1 needs to encapsulate the packet a second time, this time for delivery to B2.

13. Security Considerations

13.1. Problem Analysis

In the Softwire Mesh Framework, the data packets that are encapsulated are E-IP data packets that are traveling through the Internet. These data packets (the softwire "payload") may or may not need such security features as authentication, integrity, confidentiality, or replay protection. However, the security needs of the payload packets are independent of whether or not those packets are traversing softwires. The fact that a particular payload packet is traveling through a softwire does not in any way affect its security needs.

Thus, the only security issues we need to consider are those that affect the I-IP encapsulation headers, rather than those that affect the E-IP payload.

Since the encapsulation headers determine the routing of packets traveling through softwires, they must appear "in the clear".

In the Softwire Mesh Framework, for each receiving endpoint of a tunnel, there are one or more "valid" transmitting endpoints, where the valid transmitting endpoints are those that are authorized to tunnel packets to the receiving endpoint. If the encapsulation header has no guarantee of authentication or integrity, then it is possible to have spoofing attacks, in which unauthorized nodes send encapsulated packets to the receiving endpoint, giving the receiving endpoint the invalid impression the encapsulated packets have really traveled through the softwire. Replay attacks are also possible.

The effect of such attacks is somewhat limited, though. The receiving endpoint of a softwire decapsulates the payload and does further routing based on the IP destination address of the payload. Since the payload packets are traveling through the Internet, they have addresses from the globally unique address space (rather than, e.g., from a private address space of some sort). Therefore, these attacks cannot cause payload packets to be delivered to an address other than the one appearing in the destination IP address field of the payload packet.

However, attacks of this sort can result in policy violations. The authorized transmitting endpoint(s) of a softwire may be following a policy according to which only certain payload packets get sent through the softwire. If unauthorized nodes are able to encapsulate the payload packets so that they arrive at the receiving endpoint looking as if they arrived from authorized nodes, then the properly authorized policies have been side-stepped.

Attacks of the sort we are considering can also be used in denial-of-service attacks on the receiving tunnel endpoints. However, such attacks cannot be prevented by use of cryptographic authentication/integrity techniques, as the need to do cryptography on spoofed packets only makes the denial-of-service problem worse. (The assumption is that the cryptography mechanisms are likely to be more costly than the decapsulation/forwarding mechanisms. So if one tries to eliminate a flooding attack on the decapsulation/forwarding mechanisms by discarding packets that do not pass a cryptographic integrity test, one ends up just trading one kind of attack for another.)

This section is largely based on the security considerations section of RFC 4023, which also deals with encapsulations and tunnels.

13.2. Non-Cryptographic Techniques

If a tunnel lies entirely within a single administrative domain, then, to a certain extent, there are certain non-cryptographic techniques one can use to prevent spoofed packets from reaching a tunnel's receiving endpoint. For example, when the tunnel encapsulation is IP-based:

- The receiving endpoints of the tunnels can be given a distinct set of addresses, and those addresses can be made known to the border routers. The border routers can then filter out packets, destined to those addresses, that arrive from outside the domain.

- The transmitting endpoints of the tunnels can be given a distinct set of addresses, and those addresses can be made known to the border routers and to the receiving endpoints of the tunnels. The border routers can filter out all packets arriving from outside the domain with source addresses that are in this set, and the receiving endpoints can discard all packets that appear to be part of a softwire, but whose source addresses are not in this set.

If an MPLS-based encapsulation is used, the border routers can refuse to accept MPLS packets from outside the domain, or they can refuse to accept such MPLS packets whenever the top label corresponds to the address of a tunnel receiving endpoint.

These techniques assume that, within a domain, the network is secure enough to prevent the introduction of spoofed packets from within the domain itself. That may not always be the case. Also, these techniques can be difficult or impossible to use effectively for tunnels that are not in the same administrative domain.

A different technique is to have the encapsulation header contain a cleartext password. The 64-bit "cookie" of L2TPv3 [RFC3931] is sometimes used in this way. This can be useful within an administrative domain if it is regarded as infeasible for an attacker to spy on packets that originate in the domain and that do not leave the domain. An attacker would then not be able to discover the password. An attacker could, of course, try to guess the password, but if the password is an arbitrary 64-bit binary sequence, brute force attacks that run through all the possible passwords would be infeasible. This technique may be easier to manage than ingress filtering is, and may be just as effective if the assumptions hold. Like ingress filtering, though, it may not be applicable for tunnels that cross domain boundaries.

Therefore, it is necessary to also consider the use of cryptographic techniques for setting up the tunnels and for passing data through them.

13.3. Cryptographic Techniques

If the path between the two endpoints of a tunnel is not adequately secure, then:

- If a control protocol is used to set up the tunnels (e.g., to inform one tunnel endpoint of the IP address of the other), the control protocol MUST have an authentication mechanism, and this MUST be used when the tunnel is set up. If the tunnel is set up automatically as the result of, for example, information distributed by BGP, then the use of BGP's MD5-based authentication mechanism [RFC2385] is satisfactory.

- Data transmission through the tunnel should be secured with IPsec. In the remainder of this section, we specify the way IPsec may be used, and the implementation requirements we mention are meant to be applicable whenever IPsec is being used.

We consider only the case where IPsec is used together with an IP-based tunneling mechanism. Use of IPsec with an MPLS-based tunneling mechanism is for further study.

If it is deemed necessary to use tunnels that are protected by IPsec, the tunnel type SHOULD be negotiated by the tunnel endpoints using the procedures specified in [RFC5566]. That document allows the use of IPsec tunnel mode but also allows one to treat the tunnel head and the tunnel tail as the endpoints of a Security Association, and to use IPsec transport mode.

In order to use IPsec transport mode, encapsulated packets should be viewed as originating at the tunnel head and as being destined for the tunnel tail. A single IP address of the tunnel head will be used as the source IP address, and a single IP address of the tunnel tail will be used as the destination IP address. This technique can be used to carry MPLS packets through an IPsec Security Association, by first encapsulating the MPLS packets in MPLS-in-IP or MPLS-in-GRE [RFC4023] and then applying IPsec transport mode.

When IPsec is used to secure softwires, IPsec MUST provide authentication and integrity. Thus, the implementation MUST support either ESP (IP Encapsulating Security Payload) with null encryption [RFC4303] or else AH (IP Authentication Header) [RFC4302]. ESP with encryption MAY be supported. If ESP is used, the tunnel tail MUST check that the source IP address of any packet received on a given SA (IPsec Security Association) is the one expected, as specified in Section 5.2, step 4, of [RFC4301].

Since the softwires are set up dynamically as a byproduct of passing routing information, key distribution MUST be done automatically by means of IKEv2 [RFC4306]. If a PKI (Public Key Infrastructure) is not available, the IPsec Tunnel Authenticator sub-TLV described in [RFC5566] MUST be used and validated before setting up an SA.

The selectors associated with the SA are the source and destination addresses of the encapsulation header, along with the IP protocol number representing the encapsulation protocol being used.

14. References

14.1. Normative References

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3931] Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

[RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, March 2005.

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, April 2009.

[RFC5566] Berger, L., White, R. and E. Rosen, "BGP IPsec Tunnel Encapsulation Attribute", RFC 5566, June 2009.

[V4NLRI-V6NH] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, May 2009.

[V6NLRI-V4NH] De Clercq, J., Ooms, D., Prevost, S., and F. Le Faucheur, "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)", RFC 4798, February 2007.

14.2. Informative References

[BFD] Katz, D. and D. Ward, "Bidirectional Forwarding Detection", Work in Progress, February 2009.

[L3VPN-MCAST] Rosen, E., Ed., and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", Work in Progress, March 2009.

[L3VPN-MCAST-BGP] Aggarwal, R., Rosen, E., Morin, T. and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", Work in Progress, April 2009.

[MLDP] Minei, I., Ed., Kompella, K., Wijnands, IJ., Ed., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", Work in Progress, April 2009.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[RFC4176] El Mghazli, Y., Ed., Nadeau, T., Boucadair, M., Chan, K., and A. Gonguet, "Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management", RFC 4176, October 2005.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4306] Kaufman, C., Ed., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[RFC4378] Allan, D., Ed., and T. Nadeau, Ed., "A Framework for Multi-Protocol Label Switching (MPLS) Operations and Management (OAM)", RFC 4378, February 2006.

[RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, April 2006.

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, October 2007.

[RFC5496] Wijnands, IJ., Boers, A., and E. Rosen, "The Reverse Path Forwarding (RPF) Vector TLV", RFC 5496, March 2009.

[SW-PROB] Li, X., Ed., Dawkins, S., Ed., Ward, D., Ed., and A. Durand, Ed., "Softwire Problem Statement", RFC 4925, July 2007.

15. Contributors

Xing Li
Tsinghua University
Department of Electronic Engineering, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5983
EMail: xing@cernet.edu.cn

Simon Barber
Cisco Systems, Inc.
250 Longwater Avenue
Reading, ENGLAND, RG2 6GB
United Kingdom

EMail: sbarber@cisco.com

Pradosh Mohapatra
Cisco Systems, Inc.
3700 Cisco Way
San Jose, CA 95134
USA

EMail: pmohapat@cisco.com

John Scudder
Juniper Networks
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA

EMail: jgs@juniper.net

16. Acknowledgments

David Ward, Chris Cassar, Gargi Nalawade, Ruchi Kapoor, Pranav Mehta, Mingwei Xu, and Ke Xu provided useful input into this document.

Authors' Addresses

Jianping Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5983
EMail: jianping@cernet.edu.cn

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
EMail: yong@csnet1.cs.tsinghua.edu.cn

Chris Metz
Cisco Systems, Inc.
3700 Cisco Way
San Jose, CA 95134
USA

EMail: chmetz@cisco.com

Eric C. Rosen
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA 01719
USA

EMail: erosen@cisco.com