Network Working Group
Internet Draft
Intended status: Standards Track
Expires: September 7, 2011

A. Csaszar
G. Enyedi
S. Kini
Ericsson

March 7, 2011

IP Fast Re-Route with Fast Notification
draft-csaszar-ipfrr-fn-00.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on September 7, 2011.

Copyright Notice

Abstract

This document describes a mechanism that provides IP fast reroute (IPFRR) by using a failure notification (FN) to nodes beyond the ones that first detect the failure (i.e. nodes that are directly connected to the failure point). The paths used when IPFRR-FN is active are in most cases identical to those used after Interior Gateway Protocol (IGP) convergence. The proposed mechanism can address all single node and link failures in an area and has been designed to allow traffic recovery traffic to happen quickly (The goal being to recover in under 50msec).

Table of Contents

1. Introduction

Convergence of link-state IGPs, such as OSPF or IS-IS, after a link or node failure is known to be relatively slow. While this may be sufficient for many applications, some network SLAs and applications require faster reaction to network failures.

IGP convergence time is composed mainly of:

1. Failure detection at nodes adjacent to the failure

2. Advertisement of the topology change

3. Calculation of new routes

4. Installing new routes to linecards

Traditional Hello-based failure detection methods of link-state IGPs are relatively slow, hence a new, optimized, Hello protocol has been standardized [BFD] which can reduce failure detection times to the range of 10ms even if no lower layer notices the failure quickly (like loss of signal, etc.).

Even with fast failure detection, reaction times of IGPs may take several seconds, and even with a tuned configuration it may take at least a couple of hundreds of milliseconds.

To decrease fail-over time even further, IPFRR techniques [RFC5714], can be introduced. IPFRR solutions compliant with [RFC5714] are targeting fail-over time reduction of steps 2-4 with the following design principles:

IGP		IPFRR
2. Advertisement of the topology change	==>	No explicit advertisement, only local repair
3. Calculation of new routes	==>	Pre-computation of new routes
4. Installing new routes to linecards	==>	Pre-installation of backup routes

Pre-computing means that the way of bypassing a failed resource is computed before any failure occurs. In order to limit complexity, IPFRR techniques typically prepare for single link, single node and single Shared Risk Link Group (SRLG) failures, which failure types are undoubtedly the most common ones. The pre-calculated backup routes are also downloaded to linecards in preparation for the failure, in this way sparing the lengthy communication between control plane and data plane when a failure happens.

The principle of local rerouting requires forwarding a packet along a detour even if only the immediate neighbors of the failed resource know the failure. IPFRR methods observing the local rerouting principle do not explicitly propagate the failure information.

Unfortunately, packets on detours must be handled in a different way than normal packets as otherwise they might get returned to the failed resource. Rephrased, a node not having *any* sort of information about the failure may loop the packet back to the node from where it was rerouted - simply because its default routing/forwarding configuration dictates that. As an example, see the following figure. Assuming a link failure between A and Dst, A needs to drop packets heading to Dst. If node A forwarded packets to Src, and if the latter had absolutely no knowledge of the failure, a loop would be formed between Src and A.

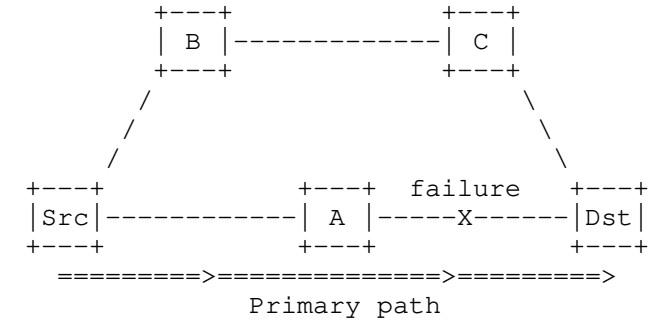


Figure 1 Forwarding inconsistency in case of local repair: The path of Src to Dst leads through A

The basic problem that previous IPFRR solutions struggle to solve is, therefore, to provide consistent routing hop-by-hop without explicit signaling of the failure.

To provide protection for all single failure cases in arbitrary topologies, the information about the failure must be given in *some* way to other nodes. That is, IPFRR solutions targeting full failure coverage need to signal the fact and to some extent the identity of the failure within the data packet as no explicit signaling is allowed. Such solutions have turned out to be considerably complex and hard or impossible to implement practically. The Loop Free Alternates (LFA) solution [RFC5286] does not give the failure information in any way to other routers, and so it cannot repair all failure cases such as the one in Figure 1.

As discussed in Section 2. , solutions that address full failure coverage and rely on local repair, i.e. carrying some failure information within the data packets, fail to present a practical alternative to LFA. This draft, therefore, suggests that relaxing the

local re-routing principle with carefully engineered explicit failure signaling is an effective approach.

The idea of using explicit failure notification for IPFRR has been proposed before for Remote LFA Paths [RLFAP]. RLFAP limits the radius in which the notification is propagated. This draft attempts to work out in more detail what kind of failure dissemination mechanism is required to facilitate remote repair efficiently. Requirements for explicit signaling are given in Section 3. This draft does not limit the failure advertisement radius as opposed to RLFAP. As a result, the detour paths remain stable in most cases, since they are identical to those that the IGP will calculation after IGP convergence. Hence, micro-loop will not occur after IGP convergence.

Note that the current -00 version of the draft only targets protection of single link and single node failures. SRLG protection is left for a future revision.

2. Overview of current IPFRR Proposals based on Local Repair

The only practically feasible solution, Loop Free Alternates [RFC5286], offers the simplest resolution of the consistency problem: a node performing fail-over may only use a next-hop as backup if it is guaranteed that it does not send the packets back. These neighbors are called Loop-Free Alternates (LFA). LFAs, however, do not always exist, as shown in Figure 1 above, i.e., node A has no LFAs with respect to Dst. while it is true that tweaking the network configuration may boost LFA failure case coverage considerably [Ret2011], LFAs cannot protect all failure cases in arbitrary network topologies.

The exact way of adding the information to data packets and its usage for forwarding is the most important property that differentiates most existing IPFRR proposals.

Packets can be marked "implicitly", when they are not altered in any way, but some extra information owned by the router helps deciding the correct way of forwarding. Such extra information can be for instance the direction of the packet, e.g., the interface, which the packet arrived through, e.g. as in [FIFR]. Such solutions require what is called interface-based or interface-specific forwarding.

Interface-based forwarding significantly changes the well-established nature of IP's destination-based forwarding principle, where the IP destination address alone describes the next hop. One embodiment would need to download different FIBs for each physical or virtual IP interface - not a very compelling idea. Another embodiment would

alter the next-hop selection process by adding the incoming interface id also to the lookup fields, which would impact forwarding performance considerably.

Other solutions mark data packets explicitly. Some proposals suggest using free bits in the IP header [MRC], which unfortunately do not exist in the IPv4 header. Other proposals resort to encapsulating re-routed packets with an additional IP header as in e.g. [NotVia] or [Eny2009b]. Encapsulation raises the problem of fragmentation and reassembly, which could be a performance bottleneck, if many packets are sent at MTU size. Another significant problem is the additional management complexity of the encapsulation addresses, which have their own semantics and need to be calculated in a failure specific manner.

3. Requirements of an Explicit Failure Signaling Mechanism

Any signaling mechanism which should be used to advertise failure notifications and so to facilitate extremely quick remote repair should have the following properties.

1. The signaling mechanism should be reliable. The mechanism needs to propagate the failure information to all interested nodes even in a network where a single link or a node is down.

2. The mechanism should be fast in the sense that getting the notification packet to remote nodes through possible multiple hops should not require (considerably) more processing at each hop than plain fast path packet forwarding.

3. The mechanism should involve simple and efficient processing to be feasible for implementation in the dataplane. This goal manifests itself in three ways: Origination of notification should very easy, e.g. creating a simple IP packet, the payload of which can be filled easily. When receiving the packet, it should be easy to recognize by dataplane linecards so that processing can commence after forwarding. No complex operations should be required in order to extract the information from the packet needed to activate the correct backup routes.

4. The mechanism should be trustable; that is, it should provide means to verify the authenticity of the notifications without significant increase of the processing burden in the dataplane.

5. Duplication of notification packets should be either strictly bounded or handled without significant dataplane processing burden.

These requirements present a trade-off. A proper balance needs to be found that offers good enough authentication and reliability while keeping processing complexity sufficiently low to be feasible for data plane implementation. One such solution is proposed in [fn-transport], which is the assumed notification protocol in the following.

4. Conceptual Operation of IPFRR relying on Fast Notification

This section outlines the operation of an IPFRR mechanism relying on Fast Notification.

4.1. Preparation Phase

Like each IPFRR solution, here it is also required to have means for quick failure detection in place, such as lower layer notifications or BFD.
The FN service needs to be activated and configured. The FN service should be bound to failure detection in such a way that FN can disseminate the information identifying the failure to the area.

Failure specific alternative path computation should typically be executed at lower priority than other routing processing.

Pre-computing the next hops on the new shortest paths for all the possible single failures may seem complex, however, it is not so difficult to realize: First, it can be done "offline", while the network is intact and the CP has few things to do. Second, for a single node, it is not needed to compute all the shortest paths with respect to any possible failures; only those link failures are needed to be taken into consideration, which are in the shortest path tree starting from the node.

After having calculated the failure specific alternative next-hops, those which represent a change to the primary next-hop, should be pre-installed to the linecards together with the identifier of the failure, which triggers the switch-over. (The resource needs of an example implementation are briefly discussed in Appendix A.)

4.2. Failure Reaction Phase

The main steps to be taken after a failure are the following:

1. Quick dataplane failure detection

2. Send information about failure using FN service right from dataplane.

3. Forward the received notification as defined by the actually used FN protocol such as the one in [fn-transport]

4. After learning about a local or remote failure, identify failure and activate failure specific backups, if needed, directly within dataplane

After a node detects the loss of connectivity to another node, it should make a decision whether the failure can be handled locally. If local repair is not possible or not configured, for example because LFA is not configured or there are destinations for which no LFA exists, it should trigger the FN service to disseminate the failure description. For instance, if BFD detects a dataplane failure it normally invokes routines to notify the control plane. For the purpose of IPFRR, BFD (or any other lower layer failure detection method) should first trigger FPN before notifying the CP.

After receiving the trigger, without any DP-CP communication involved, FN constructs a packet and adds the description of the failure (described in Section 5.1.) to the payload. The description shall enable recipient nodes to decode that, e.g., node X lost connectivity to node Z. The encoding of the IPFRR-FN packet is described in Section 5.1.

The packet is then disseminated by the FN service in the routing area. Note the synergy of the relation between BFD and IGP Hellos and between FN and IGP link state advertisements. BFD makes a dataplane optimized implementation of the routing protocol's Hello mechanism, Fast Notification makes a dataplane optimized implementation of the link state advertisement flooding mechanism of IGPs.

In each hop, the recipient node needs to perform a "punt and forward". That is, the FN packet not only needs to be forwarded to the FN neighbors as the specific FN mechanism dictates, but a replica needs to be detached and, after forwarding, started to be processed by the dataplane card.

4.2.1. Activating Failure Specific Backups

After the forwarding element extracted the contents of the notification packet, it knows that a node X has lost connectivity to a node Z via a link L. The recipient now needs to decide whether the failure was a link or a node failure. Two approaches can be thought

of. Both options are based on the property that notifications advance in the network as fast as possible.

In the first option, the router does not immediately make the decision, but instead starts a timer set to fire after a couple of milliseconds. If, the failure was a node failure, the node will receive further notifications saying that another node Y has lost connectivity to node Z through another link M. That is, if node Z is common in the notifications, the recipient can conclude that it is a node failure and already knows which node it is (Z). If link L is common in the notifications, then the recipient can decide for link failure (L). If further inconclusive notifications arrive, then it means multiple failures which case is not in scope for IPFRR, and is left for regular IGP convergence.

After concluding about the exact failure, the data plane element needs to check in its pre-installed IPFRR database whether this particular failure results in any route changes. If yes, the linecard replaces the next-hops impacted by that failure with their failure specific backups which were pre-installed in the preparation phase.

In the second option, the first received notification is handled immediately as a link failure, hence the router may start replacing its next-hops. In many cases this is a good decision. If, however, another notification arrives a couple of milliseconds later that points to a node failure, the router then needs to start replacing its next-hops again. This may cause a route flap but due to the quick dissemination mechanism the routing inconsistency is very short lived and likely takes only a couple of milliseconds.

This draft recommends that out of the several FN delivery options defined in [fn-transport], the Redundant Tree transport option is preferred, which ensures that any event can reach each node from any source with any single link or node failure present in the network area as long as theoretically possible. This also means that any node, when activating failure specific backup entries in its FIB, may assume that other nodes have been notified as well and have changed their FIBs to present consistent routing. The exception is the case of legacy nodes, see Section 5.2. for details.

5. Operation Details

5.1. Message Handling and Encoding

A failure identifier is needed that unambiguously describes the failed resource consistently among the nodes in the area. The schemantics of the identifiers are defined by the IGP used to pre-

calculate and pre-install the backup forwarding entries, e.g. OSPF or ISIS.

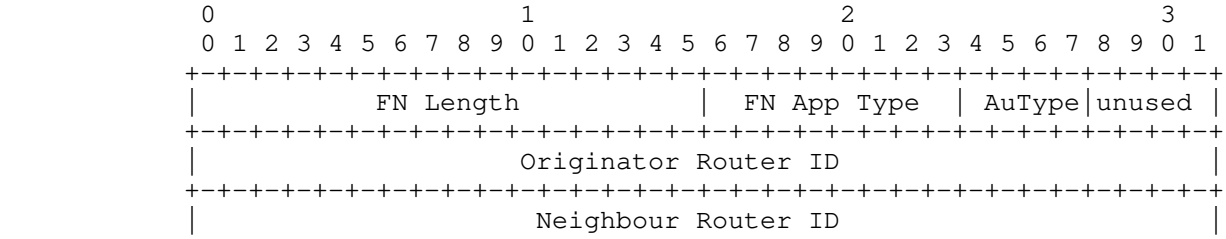
This draft defines a Failure Identification message class. Members of this class represent a routing protocol specific Failure Identification message to be carried with the Fast Notification transport protocol. Each message within the Failure Identification message class shall contain the following fields, the lengths of which are routing protocol specific. The exact values shall be aligned with the WG of the routing protocol:

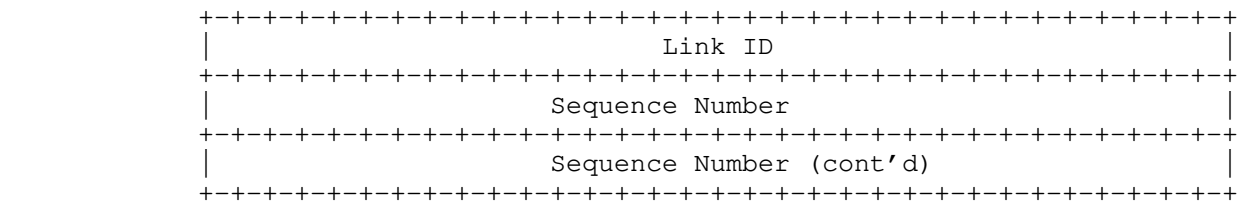
- o Originator Router ID: the identifier of the router advertising the failure;
- o Neighbour Router ID: the identifier of the neighbour node to which the originator lost connectivity.
- o Link ID: the identifier of the link, through which connectivity was lost to the neighbour. The routing protocol should assign the same Link ID for bidirectional, broadcast or multi access links from each access point, consistently.
- o Sequence Number: [fn-transport] expects the applications of the FN service that require replay attack protection to create and verify a sequence number in FN messages.

Routers forwarding the FN packets should ensure that Failure Identification messages are not lost, e.g. due to congestion. FN packets can be put a high precedence traffic class (e.g. Network Control). If the network environment is known to be lossy, the FN sender should repeat the same notification a couple of times, like a salvo fire.

After the forwarding element processed the FN packet and extracted the Failure Identification message, it should decide what backups need to be activated if at all - as described in Section 4.2.1.

5.1.1. Failure Identification Message for OSPF





FN Header fields:

FN Length
The length of the Failure Identification message for OSPF is 16 bytes.

FN App Type
The exact values are to be assigned by IANA for the Failure Identification message class. For example, FN App Type values between 0x0008 and 0x000F could represent Failure Identification messages, from which 0x0008 could mean OSPF, 0x0009 could be ISIS.

AuType
IPFRR-FN relies on the authentication options offered the FN transport service. Cryptographic authentication is recommended.

Originator Router ID
If the routing protocol is OSPF, then the value can take the OSPF Router ID of the advertising router.

Neighbour Router ID
The OSPF Router ID of the neighbour router to which connectivity was lost.

Link ID
If the link is a LAN, the Link ID takes the LSAID of its representing Network LSA.
If the link is a point-to-point link, the Link ID can take the minimum or the maximum of the two interface IDs. The requirement is that it is performed consistently.

Sequence Number
This field stores a digest of the LSDB of the routing protocol, as described in Section 6.

5.1.2. Failure Identification TLV for ISIS

TBA.

5.2. Bypassing Legacy Nodes

Legacy nodes, while cannot originate fast notifications and cannot process them either, can be assumed to be able to forward the notifications. As [fn-transport] discusses, FN forwarding is based on multicast. It is safe to assume that legacy routers' multicast configuration can be set up statically so as to be able to propagate fast notifications as needed.

When calculating failure specific alternative routes, IPFRR-FN capable nodes must consider legacy nodes as being fixed directed links since legacy nodes do not change packet forwarding in the case of failure. There are situations when an FN-IPFRR capable node can, exceptionally, bypass a non-IPFRR-FN capable node in order to handle a remote failure.

As an example consider the topology depicted in Figure 2, where the link between C and D fails. C cannot locally repair the failure.

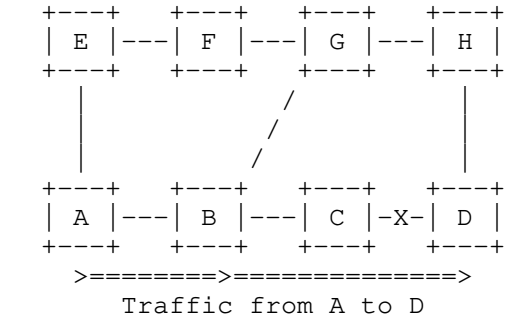


Figure 2 Example for bypassing legacy nodes

First, let us assume that each node is IPFRR-FN capable. C would advertise the failure information using FN. Each node learns that the link between C and D fails, as a result of which C changes its forwarding table to send any traffic destined to D via B. B also makes a change, replacing its default next-hop (C) with G. Note that other nodes do not need to modify their forwarding at all.

Now, let us assume that B is a legacy router not supporting IPFRR-FN but it is statically configured to multicast fast notifications as needed. As such, A will receive the notification. A's pre-

calculations have been done knowing that B is unable to correct the failure. Node A, therefore, has pre-calculated E as the failure specific next-hop. Traffic entering at A and heading to D can thus be repaired.

5.3. Capability Advertisement

The solution requires nodes to know which other nodes in the area are capable of IPFRR-FN. The most straightforward way to achieve this is to rely on the Router Capability TLVs available both in OSPF [RFC4970] and in IS-IS [RFC4971].

6. Protection against Replay Attacks

To defend against replay attacks, recipients should be able to ignore a re-sent recording of a previously sent FN packet. This suggests that some sort of sequence number should be included in the FN packet, the verification of which should not need control plane involvement. Since the solution should be simple to implement in the dataplane, maintaining and verifying per-source sequence numbers is not the best option.

We propose, therefore, that messages should be stamped with the digest of the actual routing configuration, i.e., a digest of the link state database of the link state routing protocol. The digest has to be picked carefully, so that if two LSDBs describe the same connectivity information, their digest should be identical as well, and different LSDBs should result in different digest values with high probability.

The conceptual way of handling these digests could be the following:

- When the LSDB changes, the IGP re-calculates the digest and downloads the new value to the dataplane element(s), in a secure way.
- When a FN packet is originated, the digest is put into the FN message into the Sequence Number field.
- Network nodes distribute (forward) the FN packet.
- When processing, the dataplane element first performs an authentication check of the FN packet, as described in [fn-transport].

- Finally, before processing the failure notification, the dataplane element should check whether its own known LSDB digest is identical with the one in the message.

If due to a failure event a node disseminates a failure notification with FN, an attacker might capture the whole packet and re-send it later. If it resends the packet after the IGP re-converged on the new topology, the active LSDB digest is different, so the packet can be ignored. If the packet is replayed to a recipient who still has the same LSDB digest, then it means that the original failure notification was already processed but the IGP has not yet finished converging; the IPFRR detour is already active, the replica has no impact.

6.1. Calculating LSDB Digest

We propose to create an LSDB digest that is conceptually similar to [ISISDigest]. The operation is proposed to be the following:

- Create a hash from each LSA(OSPF)/LSP(ISIS) one by one
- XOR these hashes together
- When an LSA/LSP is removed, the new LSDB digest is received by computing the hash of the removed LSA, and then XOR to the existing digest
- When an LSA/LSP is added, the new LSDB digest is received by computing the hash of the new LSA, and then XOR to the existing digest

7. Security Considerations

The IPFRR application of Fast Notification does not raise further known security consideration in addition to those already present in Fast Notification itself. If an attacker could send false Failure Identification Messages or could hinder the transmission of legal messages, then the network would produce an undesired routing behavior. These issues should be solved, however, in [fn-transport].

IPFRR-FN relies on the authentication mechanism provided by the Fast Notification transport protocol [fn-transport]. The specification of the FN transport protocol requires applications to protect against replay attacks with application specific sequence numbers. This draft, therefore, describes its own proposed sequence number in Section 6.

8. IANA Considerations

The Failure Identification message types need to be allocated a value in the FN App Type field.

IPFRR-FN capability needs to be allocated within Router Capability TLVs both for OSPF [RFC4970] and in IS-IS [RFC4971].

9. References

9.1. Normative References

[RFC5286] A. Atlas, A. Zinin, "Basic specification for IP Fast-Reroute: Loop-Free Alternates", Internet Engineering Task Force: RFC 5286, 2008.

[fn-transport] W. Lu, S. Kini, A. Csaszar, G. Enyedi, J. Tantsura, A. Tian, "Transport of Fast Notifications Messages", draft-lu-fn-transport-00, 2011

[RFC4970] A. Lindem et al., Extensions to OSPF for Advertising Optional Router Capabilities, RFC 4970, 2007

[RFC4971] JP. Vasseur et al., Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information, RFC 4971, 2007

9.2. Informative References

[BFD] D. Katz, D. Ward, "Bidirectional forwarding detection", RFC 5880, IETF, 2010

[RFC5714] M. Shand, S. Bryant, "IP Fast Reroute Framework", RFC 5714, IETF, 2010.

[Eny2009a]Gabor Enyedi, Gabor Retvari, Andras Csaszar, "On Finding Maximally Redundant Trees in Strictly Linear Time", IEEE Symposium on Computers and Communications (ISCC), 2009.

[Eny2009b] Gabor Enyedi, Peter Szilagyi, Gabor Retvari, Andras Csaszar, "IP Fast ReRoute: Lightweight Not-Via without Additional Addresses", IEEE INFOCOM-MiniConference, Rio de Janeiro, Brazil, 2009.

[FIFR] J. Wand, S. Nelakuditi, "IP fast reroute with failure inferencing", In Proceedings of ACM SIGCOMM Workshop on Internet Network Management - The Five-Nines Workshop, 2007.

[MRC] T. Cicic, A. F. Hansen, A. Kvalbein, M. Hartmann, R. Martin, M. Menth, S. Gjessing, O. Lysne, "Relaxed multiple routing configurations IP fast reroute for single and correlated failures", IEEE Transactions on Network and Service Management, available online: http://www3.informatik.uni-wuerzburg.de/staff/menth/Publications/papers/Menth08-Sub-4.pdf, September 2010.

[NotVia] S. Bryant, M. Shand, S. Previdi, "IP fast reroute using Not-via addresses", Internet Draft, draft-ietf-rtgwg-ipfrr-notvia-addresses-06, 2010.

[RLFAP] I. Hokelek, M. Fecko, P. Gurung, S. Samtani, S. Cevher, J. Sucec, "Loop-Free IP Fast Reroute Using Local and Remote LFAPs", Internet Draft, draft-hokelek-rlfap-01 (expired), 2008.

[Ret2011] G. Retvari, J. Tapolcai, G. Enyedi, A. Csaszar, "IP Fast ReRoute: Loop Free Alternates Revisited", to appear at IEEE INFOCOM 2011

[ISISDigest] J. Chiabaut and D. Fedyk. IS-IS Multicast Synchronization Digest. Available online: http://www.ieee802.org/1/files/public/docs2008/aq-fedyk-ISIS-digest-1108-v1.pdf, Nov 2008.

10. Acknowledgments

The authors would like to thank Jeff Tantsura, Albert Tian, Wenhu Lu and Acee Lindem for the continuous discussions and comments on the topic, as well as Joel Halpern for his comments and review.

This document was prepared using 2-Word-v2.0.template.dot.

Appendix A. Memory needs of a Naive Implementation

Practical background might suggest that storing and maintaining backup next-hops for many potential remote failures could overwhelm the resources of router linecards. This section attempts to provide a calculation describing the approximate memory needs in reasonable sized networks with a possible implementation.

A.1. An Example Implementation

Let us suppose that the forwarding engine is optimized for forwarding performance in the sense that recursive lookups are not performed for external destinations but each IP lookup gives back an adjacency (a number describing the next hop for the router), even if the packet will be terminated outside the current area. From the aspect of storing backup next-hops per destination, this is worse than using recursive lookup, since in this case the update of a lot more destinations is needed. In case of recursive lookup, external prefixes are resolved to internal destinations, so we can simply choose not to deal with external prefixes.

This implementation uses an array for all the nodes in the area (node array in the sequel), made up by two pointers per record. Both of these pointers point to another array with a header describing its lengths. The first array (called alternative array) is basically an enumeration containing the IDs of those failures influencing a shortest path towards that node and an alternative neighbor, which can be used, when such a failure occurs. When a failure is detected, (either locally, or by FN), we can easily find the proper record in all the lists. Moreover, since these arrays can be sorted based on the failure ID, we can even use binary search to find the needed record.

Now, we only need to know, which records in the FIB should be updated. Therefore there is a second pointer in the node array pointing to another enumeration (called FIB array in the sequel) containing pointers to the corresponding FIB entries. Recall, that if the node is an egress router, FIB array contains more than one entry. Moreover, there can be some prefixes reachable through more than one egres routers, thus these entries may be in more than one FIB arrays.

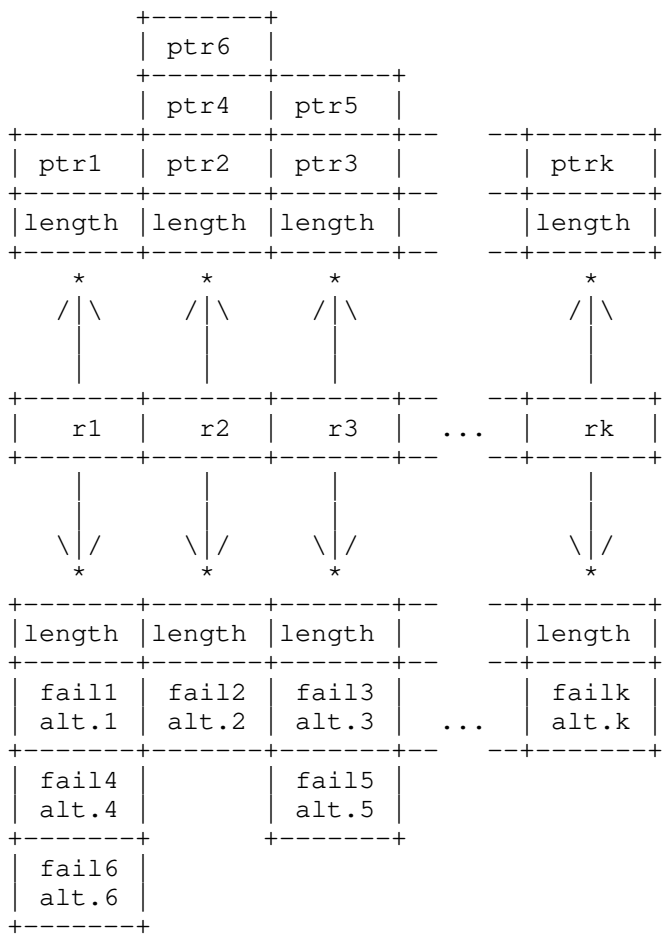


Figure 3 The way of storing alternatives

A.2. Estimation of Memory Requirements.

Now, suppose that there are D prefixes, the area containing the router has V routers, a prefix is connected to K egress routers in average, a neighbor descriptor takes X bytes, a failure ID takes Y bytes and a pointer takes Z bytes. In this way, if there is no ECMP, this data structure takes

$$2*Z*(V-1) + (2*(X+Y)*(V-1)+Y)*(V-1) + (K*D*Z+Y*(V-1))$$

bytes altogether. The first part is the memory consumption of the node array. The memory consumption of all the FIB arrays is described

by the last part (V-1 length fields and K*D pointers). The remaining part describes the maximum memory needed by an alternative arrays: any path can contain at most V-1 nodes and V-1 links, each record needs X+Y bytes plus we have a header for the array; there are records for all the other nodes in the area (V-1 nodes). Observe that this is a very rough overestimation, since most of the possible failures influencing the path will not change the next hop.

For computing memory consumption, suppose that neighbor descriptors, failure IDs and pointers take 4 bytes, there are 200 (500) nodes in the area and we have 500K prefixes installed, and a prefix is reachable through 2 egress routers in average. In this case, we get that the node array needs about 1.6KB (4KB), the alternative array needs about 620KB (4MB), and the FIB array needs about 4MB (4MB). That is altogether less than 5MB (8MB) in reality, if there is no ECMP.

If however, there are paths with equal costs, the size of the alternative array increases. Suppose that there are 10 equal paths between ANY two nodes in the network. This would cause that the alternative list gets 10 times bigger, and now it needs 6.2MB. Observe that now we need about 11MB (44MB) even in this extremely unrealistic case, which is likely acceptable for modern linecards with gigs of DRAM. Moreover, we need to stress here again that this is an extremely rough overestimation, so in reality much less memory will be enough.

Authors' Addresses

Andras Csaszar
Ericsson
Irinyi utca 4-10, Budapest, Hungary, 1117
Email: Andras.Csaszar@ericsson.com

Gabor Sandor Enyedi
Ericsson
Irinyi utca 4-10, Budapest, Hungary, 1117
Email: Gabor.Sandor.Enyedi@ericsson.com

Sriganesh Kini
Ericsson
300 Holger Way, San Jose, CA 95134
Email: sriganesh.kini@ericsson.com

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: December 6, 2012

A. Csaszar (Ed.)
G. Enyedi
J. Tantsura
S. Kini
Ericsson

J. Sucec
S. Das
Telcordia

June 6, 2012

IP Fast Re-Route with Fast Notification
draft-csaszar-ipfrr-fn-03.txt

Status of this Memo

Copyright Notice

carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document describes the benefits and main applications of sending explicit fast notification (FN) packets to routers in an area. FN packets are generated and processed in the dataplane, and a single FN service can substitute existing OAM methods for remote failure detection, such as a full mesh of multi-hop BFD session. The FN service, therefore, decreases network overhead considerable. The main application is fast reroute in pure IP and in IP/LDP-MPLS networks called IPFRR-FN. The detour paths used when IPFRR-FN is active are in most cases identical to those used after Interior Gateway Protocol (IGP) convergence. The proposed mechanism can address all single link, node, and SRLG failures in an area; moreover it is an efficient solution to protect against BGP ASBR failures as well as VPN PE router failures. IPFRR-FN can be a supplemental tool to provide FRR when LFA cannot repair a failure case, while it can be a replacement of existing ASBR/PE protection mechanisms by overcoming their scalability and complexity issues.

Table of Contents

1. Introduction

Convergence of link-state IGPs, such as OSPF or IS-IS, after a link or node failure is known to be relatively slow. While this may be sufficient for many applications, some network SLAs and applications require faster reaction to network failures.

IGP convergence time is composed mainly of:

1. Failure detection at nodes adjacent to the failure

2. Advertisement of the topology change

3. Calculation of new routes

4. Installing new routes to linecards

Traditional Hello-based failure detection methods of link-state IGPs are relatively slow, hence a new, optimized, Hello protocol has been standardized [BFD] which can reduce failure detection times to the range of 10ms even if no lower layer notices the failure quickly (like loss of signal, etc.).

Even with fast failure detection, reaction times of IGPs may take several seconds, and even with a tuned configuration it may take at least a couple of hundreds of milliseconds.

To decrease fail-over time even further, IPFRR techniques [RFC5714], can be introduced. IPFRR solutions compliant with [RFC5714] are targeting fail-over time reduction of steps 2-4 with the following design principles:

IGP		IPFRR
2. Advertisement of the topology change	==>	No explicit advertisement, only local repair
3. Calculation of new routes	==>	Pre-computation of new routes
4. Installing new routes to linecards	==>	Pre-installation of backup routes

Pre-computing means that the way of bypassing a failed resource is computed before any failure occurs. In order to limit complexity, IPFRR techniques typically prepare for single link, single node and single Shared Risk Link Group (SRLG) failures, which failure types are undoubtedly the most common ones. The pre-calculated backup routes are also downloaded to linecards in preparation for the failure, in this way sparing the lengthy communication between control plane and data plane when a failure happens.

The principle of local rerouting requires forwarding a packet along a detour even if only the immediate neighbors of the failed resource know the failure. IPFRR methods observing the local rerouting principle do not explicitly propagate the failure information. Unfortunately, packets on detours must be handled in a different way than normal packets as otherwise they might get returned to the failed resource. Rephrased, a node not having *any* sort of information about the failure may loop the packet back to the node from where it was rerouted - simply because its default routing/forwarding configuration dictates that. As an example, see the following figure. Assuming a link failure between A and Dst, A needs to drop packets heading to Dst. If node A forwarded packets to Src, and if the latter had absolutely no knowledge of the failure, a loop would be formed between Src and A.

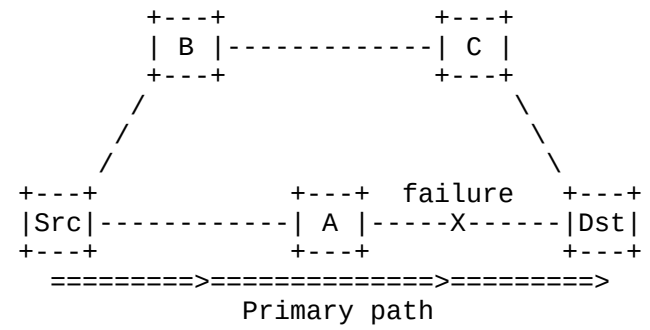


Figure 1 Forwarding inconsistency in case of local repair: The path of Src to Dst leads through A

The basic problem that previous IPFRR solutions struggle to solve is, therefore, to provide consistent routing hop-by-hop without explicit signaling of the failure.

To provide protection for all single failure cases in arbitrary topologies, the information about the failure must be given in *some* way to other nodes. That is, IPFRR solutions targeting full failure coverage need to signal the fact and to some extent the identity of the failure within the data packet as no explicit signaling is allowed. Such solutions have turned out to be considerably complex and hard or impossible to implement practically. The Loop Free Alternates (LFA) solution [RFC5286] does not give the failure information in any way to other routers, and so it cannot repair all failure cases such as the one in Figure 1.

As discussed in Section 2. solutions that address full failure coverage and rely on local repair, i.e. carrying some failure information within the data packets, present an overly complex and therefore often inpractical alternative to LFA. This draft, therefore, suggests that relaxing the local re-routing principle with carefully engineered explicit failure signaling is an effective approach.

The idea of using explicit failure notification for IPFRR has been proposed before for Remote LFA Paths [RLFAP]. RLFAP sends explicit notifications and can limit the radius in which the notification is propagated to enhance scalability. Design, implementation and enhancements for the remote LFAP concept are reported in [Hok2007], [Hok2008] and [Cev2010].

This draft attempts to work out in more detail what kind of failure dissemination mechanism is required to facilitate remote repair efficiently. Requirements for explicit signaling are given in Section 3. This draft does not limit the failure advertisement radius as opposed to RLFAP. As a result, the detour paths remain stable in most cases, since they are identical to those that the IGP will calculation after IGP convergence. Hence, micro-loop will not occur after IGP convergence.

A key contribution of this memo is to recognize that a Fast Notification service is not only an enabler for a new IPFRR approach but it is also a replacement for various OAM remote connectivity verification procedures such as multi-hop BFD. These previous methods posed considerable overhead to the network: (i) management of many OAM sessions; (ii) careful configuration of connectivity verification packet interval so that no false alarm is given for network internal failures which are handled by other mechanisms; and (iii) packet processing overhead, since connectivity verification packets have to be transmitted continuously through the network in a mesh, even in fault-free conditions.

2. Overview of current IPFRR Proposals based on Local Repair

The only practically feasible solution, Loop Free Alternates [RFC5286], offers the simplest resolution of the hop-by-hop routing consistency problem: a node performing fail-over may only use a next-hop as backup if it is guaranteed that it does not send the packets back. These neighbors are called Loop-Free Alternates (LFA). LFAs, however, do not always exist, as shown in Figure 1 above, i.e., node A has no LFAs with respect to Dst. while it is true that tweaking the network configuration may boost LFA failure case coverage considerably [Ret2011], LFAs cannot protect all failure cases in arbitrary network topologies.

The exact way of adding extra information to data packets and its usage for forwarding is the most important property that differentiates most existing IPFRR proposals.

Packets can be marked "implicitly", when they are not altered in any way, but some extra information owned by the router helps deciding the correct way of forwarding. Such extra information can be for instance the direction of the packet, e.g., the incoming interface, e.g. as in [FIFR]. Such solutions require what is called interface-based or interface-specific forwarding.

Interface-based forwarding significantly changes the well-established nature of IP’s destination-based forwarding principle, where the IP

destination address alone describes the next hop. One embodiment would need to download different FIBs for each physical or virtual IP interface - not a very compelling idea. Another embodiment would alter the next-hop selection process by adding the incoming interface id also to the lookup fields, which would impact forwarding performance considerably.

Other solutions mark data packets explicitly. Some proposals suggest using free bits in the IP header [MRC], which unfortunately do not exist in the IPv4 header. Other proposals resort to encapsulating re-routed packets with an additional IP header as in e.g. [NotVia], [Eny2009] or [MRT-ARCH]. Encapsulation raises the problem of fragmentation and reassembly, which could be a performance bottleneck, if many packets are sent at MTU size. Another significant problem is the additional management complexity of the encapsulation addresses, which have their own semantics and require cumbersome routing calculations, see e.g. [MRT-ALG]. Encapsulation in the IP header translates to label stacking in LDP-MPLS. The above mentioned mechanisms either encode the active topology ID in a label on the stack or encode the failure point in a label, and also require an increasing mesh of targeted LDP sessions to acquire a valid label at the detour endpoint, which is another level of complexity.

3. Requirements of an Explicit Failure Signaling Mechanism

All local repair mechanisms touched above try to avoid explicit notification of the failure via signaling, and instead try to hack some failure-related information into data packets. This is mainly due to relatively low signaling performance of legacy hardware. Failure notification, therefore, should fulfill the following properties to be practically feasible:

1. The signaling mechanism should be reliable. The mechanism needs to propagate the failure information to all interested nodes even in a network where a single link or a node is down.

2. The mechanism should be fast in the sense that getting the notification packet to remote nodes through possible multiple hops should not require (considerably) more processing at each hop than plain fast path packet forwarding.

3. The mechanism should involve simple and efficient processing to be feasible for implementation in the dataplane. This goal manifests itself in three ways:

    a. Origination of notification should be very easy, e.g. creating a simple IP packet, the payload of which can be filled easily.

   b. When receiving the packet, it should be easy to recognize by dataplane linecards so that processing can commence after forwarding.

   c. No complex operations should be required in order to extract the information from the packet needed to activate the correct backup routes.

4. The mechanism should be trustable; that is, it should provide means to verify the authenticity of the notifications without significant increase of the processing burden in the dataplane.

5. Duplication of notification packets should be either strictly bounded or handled without significant dataplane processing burden.

These requirements present a trade-off. A proper balance needs to be found that offers good enough authentication and reliability while keeping processing complexity sufficiently low to be feasible for data plane implementation. One such solution is proposed in [fn-transport], which is the assumed notification protocol in the following.

4. Conceptual Operation of IPFRR relying on Fast Notification

This section outlines the operation of an IPFRR mechanism relying on Fast Notification.

4.1. Preparation Phase

As any other IPFRR solution, IPFRR-FN also requires quick failure detection mechanisms in place, such as lower layer upcalls or BFD. The FN service needs to be activated and configured so that FN disseminates the information identifying the failure to the area once triggered by a local failure detection method.

Based on the detailed topology database obtained by a link state IGP, the node should pre-calculate alternative paths considering *relevant* link or node failures in the area. Failure specific alternative path computation should typically be executed at lower priority than other routing processing. Note that the calculation can be done "offline", while the network is intact and the CP has few things to do.

Also note the word *relevant* above: a node does not needed to compute all the shortest paths with respect to each possible failure;

only those link failures need to be taken into consideration, which are in the shortest path tree starting from the node.

To provide protection for Autonomous System Border Router (ASBR) failures, the node will need information not only from the IGP but also from BGP. This is described in detail in Section 5.3.

After calculating the failure specific alternative next-hops, only those which represent a change to the primary next-hop, should be pre-installed to the linecards together with the identifier of the failure, which triggers the switch-over. In order to preserve scalability, external prefixes are handled through FIB indirection available in most routers already. Due to indirection, backup routes need to be installed only for egress routers. (The resource needs of an example implementation are briefly discussed in Appendix A.)

4.2. Failure Reaction Phase

The main steps to be taken after a failure are the following:

1. Quick dataplane failure detection

2. Send information about failure using FN service right from dataplane.

3. Forward the received notification as defined by the actually used FN protocol such as the one in [fn-transport]

4. After learning about a local or remote failure, extract failure identifier and activate failure specific backups, if needed, directly within dataplane

5. Start forwarding data traffic using the updated FIB

After a node detects the loss of connectivity to another node, it should make a decision whether the failure can be handled locally. If local repair is not possible or not configured, for example because LFA is not configured or there are destinations for which no LFA exists, a failure should trigger the FN service to disseminate the failure description. For instance, if BFD detects a dataplane failure it not only should invoke routines to notify the control plane but it should first trigger FN before notifying the CP.

After receiving the trigger, without any DP-CP communication involved, FN constructs a packet and adds the description of the failure (described in Section 5.1.) to the payload. The notification describes that

- a node X has lost connectivity
- to a node Z
- via a link L.

The proposed encoding of the IPFRR-FN packet is described in Section 5.1.

The packet is then disseminated by the FN service in the routing area. Note the synergy of the relation between BFD and IGP Hellos and between FN and IGP link state advertisements. BFD makes a dataplane optimized implementation of the routing protocol's Hello mechanism, while Fast Notification makes a dataplane optimized implementation of the link state advertisement flooding mechanism of IGPs.

In each hop, the recipient node needs to perform a "punt and forward". That is, the FN packet not only needs to be forwarded to the FN neighbors as the specific FN mechanism dictates, but a replica needs to be detached and, after forwarding, started to be processed by the dataplane card.

4.2.1. Activating Failure Specific Backups

After the forwarding element extracted the contents of the notification packet, it knows that a node X has lost connectivity to a node Z via a link L. The recipient now needs to decide whether the failure was a link or a node failure. Two approaches can be thought of. Both options are based on the property that notifications advance in the network as fast as possible.

In the first option, the router does not immediately make the decision, but instead starts a timer set to fire after a couple of milliseconds. If, the failure was a node failure, the node will receive further notifications saying that another node Y has lost connectivity to node Z through another link M. That is, if node Z is common in multiple notifications, the recipient can conclude that it is a node failure and already knows which node it is (Z). If link L is common, then the recipient can decide for link failure (L). If further inconclusive notifications arrive, then it means multiple failures which case is not in scope for IPFRR, and is left for regular IGP convergence.

After concluding about the exact failure, the data plane element needs to check in its pre-installed IPFRR database whether this particular failure results in any route changes. If yes, the linecard

replaces the next-hops impacted by that failure with their failure specific backups which were pre-installed in the preparation phase.

In the second option, the first received notification is handled immediately as a link failure, hence the router may start replacing its next-hops. In many cases this is a good decision, as it has been shown before that most network failures are link failures. If, however, another notification arrives a couple of milliseconds later that points to a node failure, the router then needs to start replacing its next-hops again. This may cause a route flap but due to the quick dissemination mechanism the routing inconsistency is very short lived and likely takes only a couple of milliseconds.

4.2.2. SRLG Handling

The above conceptual solution is easily extensible to support pre-configured SRLGs. Namely, if the failed link is part of an SRLG, then the disseminated link ID should identify the SRLG itself. As a result, possible notifications describing other link failures of the same SRLG will identify the same resource.

If the control plane knows about SRLGs, it can prepare for failures of these, e.g. by calculating a path that avoids all links in that SRLG. SRLG identifier may have been pre-configured or have been obtained by automated mechanisms such as [RFC4203].

4.3. Example and Timing

The main message of this section is that big delay links do not represent a problem for IPFRR-FN. The FN message of course propagates on long-haul links slower but the same delay is incurred by normal data packets as well. Packet loss only takes place as long as a node forwards traffic to an incorrect or inconsistent next-hop. This may happen in two cases:

First, as long as the failure is not detected, the node adjacent to the failure only has the failed next-hop installed.

Secondly, when a node (A) selects a new next-hop (B) after detecting the failure locally or by receiving an FN, the question is if the routing in the new next-hop (B) is consistent by the time the first data packets get from A to B. The following timeline depicts the situation:

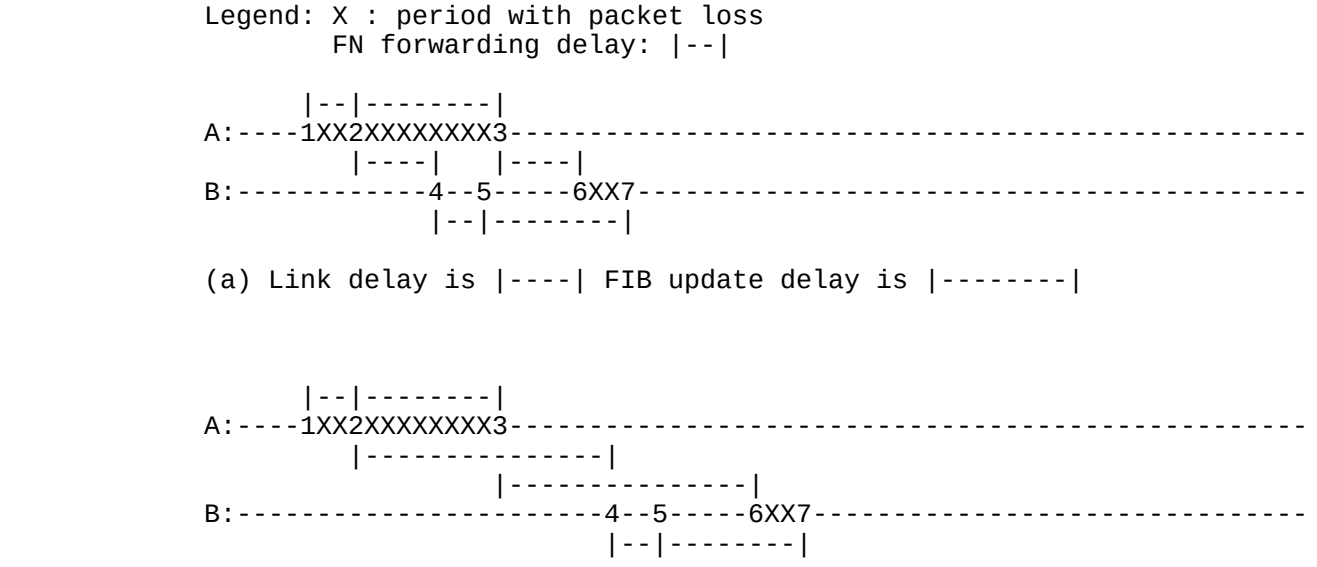


Figure 2 Timing of FN and data packet forwarding

As can be seen above, the outage time is only influenced by the FN forwarding delay and the FIB update time. The link delay is not a factor. Node A forwards the first re-routed packets from time instance 3 to node B. These reach node B at time instance 6. Node B is doing incorrect/inconsistent forwarding when it tries to forward those packets back to A which have already been put onto a detour by A. This is the interval between time instances 6 and 7.

4.4. Scoping FN Messages with TTL

In a large routing area it is often the case that a failure (i.e. a topology change) causes next-hop changes only in routers relatively close to the failure. Analysis of certain random topologies and two example ISP topologies revealed that a single link failure event generated routing table changes only in routers not more than 2 hops away from the failure site for the particular topologies under study [Hok2008]. Based on this analysis, it is anticipated that in practice the TTL for failure notification messages can be set to a relatively small radius, perhaps as small as 2 or 3 hops.

A chief benefit of TTL scoping is that it reduces the overhead on routers that have no use for the information (i.e. which do not need to re-route). Another benefit (that is particularly important for

links with scarce capacity) is that it helps to constrain the control overhead incurred on network links. Determining a suitable TTL value for each locally originated event and controlling failure notification dissemination, in general, is discussed further in Section 5.8.

5. Operation Details

5.1. Transport of Fast Notification Messages

This draft recommends that out of the several FN delivery options defined in [fn-transport], the flooding transport option is preferred, which ensures that any event can reach each node from any source with any failure present in the network area as long as theoretically possible. Flooding also ensures that FN messages reach each node on the shortest (delay) path, and as a side effect failure notifications always reach *each* node *before* re-routed data packets could reach that node. This means that looping is minimized.

[fn-transport] describes that the dataplane flooding procedure requires routers to perform duplicate checking before forwarding the notifications to other interfaces to avoid duplicating notifications. [fn-transport] describes that duplicate check can be performed by a simple storage queue, where previously received notification packets or their signatures are stored.

IPFRR-FN enables another duplicate check process that is based on the internal state machine. Routers, after receiving a notification but before forwarding it to other peers, check the authenticity of the message, if authentication is used. Now the router may check what is the stored event and what is the event described by the received notification.

Two variables and a bit describe what is the known failure state:

- Suspected failed node ID (denoted by N)
- Suspected link/SRLG ID (denoted by S)
- Bit indicating the type of the failure, i.e. link/SRLG failure or node failure (denoted by T)

Recall that the incoming notification describes that a node X has lost connectivity to a node Z via a link L. Now, the state machine can be described with the following pseudo-code:

```
//current state:
//  N: ID of suspected failed node
//  S: ID of suspected failed link/SRLG
//  T: bit indicating the type of the failure
//     T=0 indicates link/SRLG
//     T=1 indicates node
//
Proc notification_received(Node Originator_X, Node Y, SRLG L) {
    if (N == NULL) {
        // this is a new event, store it and forward it
        N=Y;
        S=L;
        T=0; //which is the default anyway
        Forward_notification;
    }
    else if (S == L AND T == 0) {
        // this is the same link or SRLG as before, need not do
        // anything
        Discard_notification;
    }
    else if (N == Y) {
        // This is a node failure
        if (T == 0) {
            // Just now turned out that it is a node failure
            T=1;
            Forward_notification;
        }
        else {
            // Known before that it is a node failure,
            // no need to forward it
            Discard_notification;
        }
    }
    else {
        // multiple failures
    }
}
```

Figure 3 Pseudo-code of state machine for FN forwarding

5.2. Message Handling and Encoding

A failure identifier is needed that unambiguously describes the failed resource consistently among the nodes in the area. The schemantics of the identifiers are defined by the IGP used to pre-calculate and pre-install the backup forwarding entries, e.g. OSPF or ISIS.

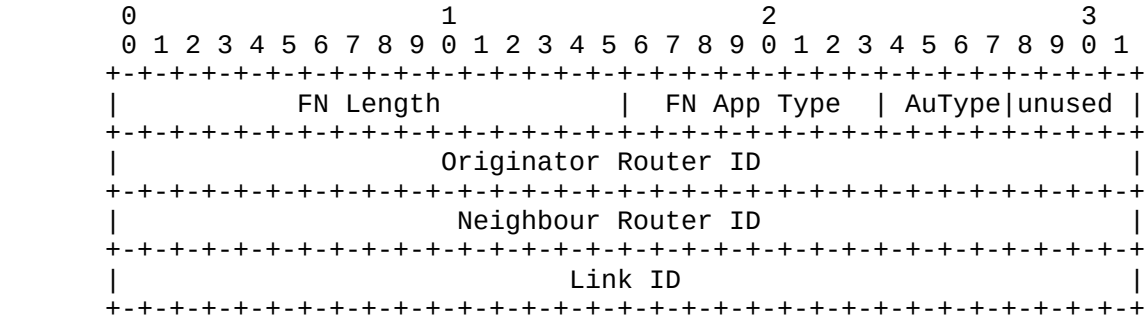
This draft defines a Failure Identification message class. Members of this class represent a routing protocol specific Failure Identification message to be carried with the Fast Notification transport protocol. Each message within the Failure Identification message class shall contain the following fields, the lengths of which are routing protocol specific. The exact values shall be aligned with the WG of the routing protocol:

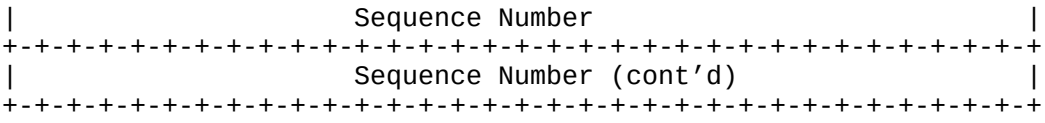
- Originator Router ID: the identifier of the router advertising the failure;
- Neighbour Router ID: the identifier of the neighbour node to which the originator lost connectivity.
- Link ID: the identifier of the link, through which connectivity was lost to the neighbour. The routing protocol should assign the same Link ID for bidirectional, broadcast or multi access links from each access point, consistently.
- Sequence Number: [fn-transport] expects the applications of the FN service that require replay attack protection to create and verify a sequence number in FN messages. It is described in Section 6.

Routers forwarding the FN packets should ensure that Failure Identification messages are not lost, e.g. due to congestion. FN packets can be put a high precedence traffic class (e.g. Network Control class). If the network environment is known to be lossy, the FN sender should repeat the same notification a couple of times, like a salvo fire.

After the forwarding element processed the FN packet and extracted the Failure Identification message, it should decide what backups need to be activated if at all - as described in Section 4.2.1.

5.2.1. Failure Identification Message for OSPF





FN Header fields:

FN Length
The length of the Failure Identification message for OSPF is 16 bytes.

FN App Type
The exact values are to be assigned by IANA for the Failure Identification message class. For example, FN App Type values between 0x0008 and 0x000F could represent Failure Identification messages, from which 0x0008 could mean OSPF, 0x0009 could be ISIS.

AuType
IPFRR-FN relies on the authentication options offered the FN transport service. Cryptographic authentication is recommended.

Originator Router ID
If the routing protocol is OSPF, then the value can take the OSPF Router ID of the advertising router.

Neighbour Router ID
The OSPF Router ID of the neighbour router to which connectivity was lost.

Link ID
If the link is a LAN, the Link ID takes the LSAID of its representing Network LSA.
If the link is a point-to-point link, the Link ID can take the minimum or the maximum of the two interface IDs. The requirement is that it is performed consistently.

Sequence Number
This field stores a digest of the LSDB of the routing protocol, as described in Section 6. 5.8.1.

5.2.2. Failure Identification Message for ISIS

TBA.

5.3. Protecting External Prefixes

5.3.1. Failure on the Intra-Area Path Leading to the ASBR

Installing failure specific backup next-hops for each external prefix would be a scalability problem as the number of these prefixes may be one or two orders of magnitude higher than intra-area destinations. To avoid this, it is suggested to make use of indirection already offered by router vendors.

Indirection means that when a packet needs to be forwarded to an external destination, the IP address lookup in the FIB will not return a direct result but a pointer to another FIB entry, i.e. to the FIB entry of the ASBR. In LDP/MPLS this means that all prefixes reachable through the same ASBR constitute the same FEC.

As an example, consider that in an area ASBR1 is the primary BGP route for prefixes P1, P2, P3 and P4 and ASBR2 is the primary route for prefixes P5, P6 and P7. A FIB arrangement for this scenario could be the one shown on the following figure. Prefixes using the same ASBR could be resolved to the same pointer that references to the next-hop leading to the ASBR. Prefixes resolved to the same pointer are said to be part of the same "prefix group" or FEC.

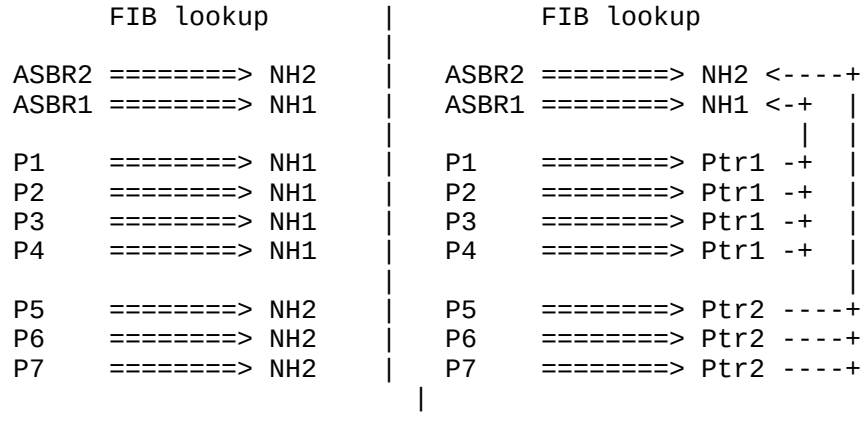


Figure 4 FIB without (left) and with (right) indirection

If the next-hop to an ASBR changes, it is enough to update in the FIB the next-hop of the ASBR route. In the above example, this means that if the next-hop of ASBR1 changes, it is enough to update the route entry for ASBR1 and due to indirection through pointer Ptr1 this updates several prefixes at the same time.

5.3.2. Protecting ASBR Failures: BGP-FRR

IPFRR-FN can make use of alternative BGP routes advertised in an AS by new extensions of BGP such as [BGPAddPaths], [DiverseBGP] or [BGPBestExt]. Using these extensions, for each destination prefix, a node may learn a "backup" ASBR besides the primary ASBR learnt by normal BGP operation.

5.3.2.1. Primary and Backup ASBR in the Same Area

If the failed ASBR is inside the area, all nodes within that area get notified by FN. Grouping prefixes into FECs, however, needs to be done carefully. Prefixes now constitute a common group (i.e. are resolved to the same pointer) if *both* their primary AND their backup ASBRs are the same. This is due to the fact that even if two prefixes use the ASBR by default, they may use different ASBRs when their common default ASBR fails.

Considering the previous example, let us assume that the backup ASBR of prefixes P1 and P2 is ASBR3 but that the backup ASBR of P3 and P4 is an ASBR2. Let us further assume that P5 also has ASBR3 as its backup ASBR but P6 and P7 have an ASBR 4 as their backup ASBR. The resulting FIB structure is shown in the following figure:

```
      FIB lookup
ASBR4 ========> NH4
ASBR2 ========> NH2
ASBR3 ========> NH3
ASBR1 ========> NH1

P1    ========> Ptr1 -+-> NH1
P2    ========> Ptr1 -+

P3    ========> Ptr2 -+-> NH1
P4    ========> Ptr2 -+

P5    ========> Ptr3 ---> NH2

P6    ========> Ptr4 -+-> NH2
P7    ========> Ptr4 -+
```

Figure 5 Indirect FIB for ASBR protection

If, for example, ASBR1 goes down, this affects prefixes P1 through P4. In order to set the correct backup routes, the container referenced by Ptr1 needs to be updated to NH2 (next-hop of ASBR2) but

the location referenced by Ptr2 needs to be updated to NH3 (next-hop of ASBR3). This means that P1 and P2 may constitute the same FEC but P3 and P4 needs to be another FEC so that there backups can be set independently.

Note that the routes towards ASBR2 or ASBR3 may have changed, too. For example, if after the failure ASBR3 would use a new next-hop NH5, then the container referenced by Ptr2 should be updated to NH5. A resulting detour FIB is shown in the following figure.

```
      FIB lookup
ASBR4 ========>    NH4
ASBR2 ========>    NH2
ASBR3 ========>    NH5
ASBR1 ========>     X

P1    ========> Ptr1 -+-> NH2
P2    ========> Ptr1 -+

P3    ========> Ptr2 -+-> NH5
P4    ========> Ptr2 -+

P5    ========> Ptr3 ---> NH2

P6    ========> Ptr4 -+-> NH2
P7    ========> Ptr4 -+
```

Figure 6 Indirect "detour" FIB in case of ASBR1 failure

During pre-calculation, the control plane pre-downloaded the failure identifier of ASBR1 and assigned NH5 as the failure specific backup for routes for ASBR3 and pointer Ptr2 and assigned NH2 as the failure specific backup for the route referenced by Ptr1.

5.3.2.2. Primary and Backup ASBR in Different Areas

By default, the scope of FN messages is limited to a single routing area.

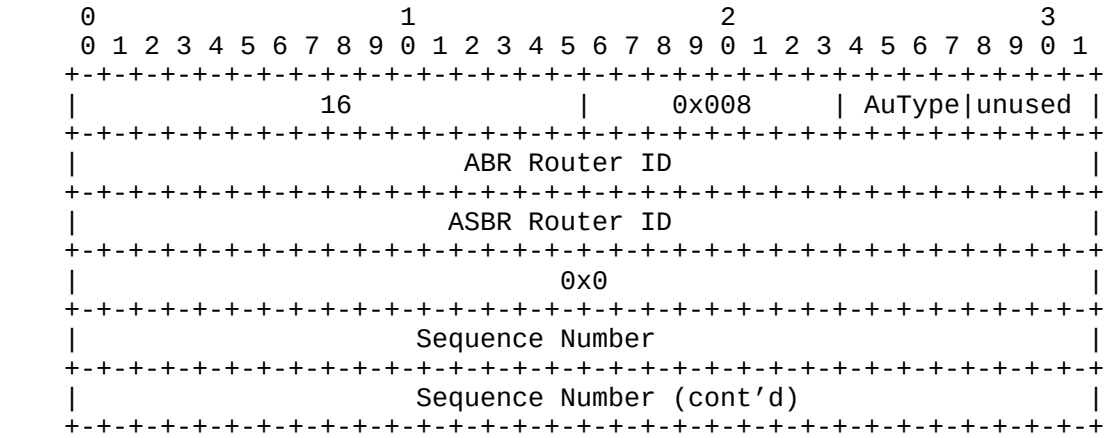
The IPFRR-FN application of FN, may, however, need to redistribute some specific notifications across areas in a limited manner.

If an ASBR1 in Area1 goes down and some prefixes need to use ASBR2 in another Area2, then, besides Area1, routers in Area2 need to know about this failure. Since communication between non-backbone areas is done through the backbone areas, it may also need the information.

Naturally, if ASBR2 resides in the backbone area, then the FN of ASBR1 failure needs to be leaked only to the backbone area.

Leaking is facilitated by area border routers (ABR). During failure preparation phase, the routing engine of an ABR can determine that for an intra-area ASBR the backup ASBR is in a different area to which it is the ABR. Therefore, the routing engine installs such intra-area ASBRs in an "FN redistribution list" at the dataplane cards.

The ABR, after receiving FN messages, may conclude in its state machine that a node failure happened. If this node failure is in the redistribution list, the ABR will generate an FN with the following data:



This message is then distributed to the neighbour area specified in the redistribution list as a regular FN message. A Link ID of 0x0 specifically signals in the neighbour area that this failure is a known node failure of the node specified by the "Neighbour Router ID" field (which was set to the failed ASBR's ID).

ABRs in a non-backbone area need to prepare to redistribute ASBR failure notifications from within their area to the backbone area.

ABRs in the backbone area need to prepare to redistribute an ASBR failure notification from the backbone area to that area where a backup ASBR resides.

Consider the previous example, but now let us assume that the current area is Area0, ASBR2 and ASBR3 reside in Area1 (reachable through ABR1) but ASBR 4 resides in Area2 (reachable through ABR2). The

resulting FIBs are shown in the following figures: in case of ASBR2 failure, only Ptr4 needs an update.

```
    FIB lookup
ABR1 ========> NH6
ABR2 ========> NH7

(ASBR4 ========> NH7)  //may or may not be in the FIB
(ASBR2 ========> NH6)  //may or may not be in the FIB
(ASBR3 ========> NH6)  //may or may not be in the FIB
(ASBR1 ========> NH1)  //may or may not be in the FIB

P1   ========> Ptr1 -+-> NH1
P2   ========> Ptr1 -+

P3   ========> Ptr2 -+-> NH1
P4   ========> Ptr2 -+

P5   ========> Ptr3 ---> NH6

P6   ========> Ptr4 -+-> NH6
P7   ========> Ptr4 -+
```

Figure 7 Indirect FIB for inter-area ASBR protection

```
    FIB lookup
ABR1 ========>    NH6
ABR2 ========>    NH7

(ASBR4  =======> NH7)  //may or may not be in the FIB
(ASBR2  =======>  X )  //may or may not be in the FIB
(ASBR3 ========> NH6)  //may or may not be in the FIB
(ASBR1 ========> NH1)  //may or may not be in the FIB

P1   ========> Ptr1 -+-> NH1
P2   ========> Ptr1 -+

P3   ========> Ptr2 -+-> NH1
P4   ========> Ptr2 -+

P5   ========> Ptr3 ---> NH6

P6   ========> Ptr4 -+-> NH7
P7   ========> Ptr4 -+
```

Figure 8 Indirect "detour" FIB for inter-area ASBR protection, ASBR2 failure

5.4. Application to LDP

It is possible for LDP traffic to follow paths other than those indicated by the IGP. To do so, it is necessary for LDP to have the appropriate labels available for the alternate so that the appropriate out-segments can be installed in the forwarding plane before the failure occurs.

This means that a Label Switching Router (LSR) running LDP must distribute its labels for the Forwarding Equivalence Classes (FECs) it can provide to all its neighbours, regardless of whether or not they are upstream. Additionally, LDP must be acting in liberal label retention mode so that the labels that correspond to neighbours that aren’t currently the primary neighbour are stored. Similarly, LDP should be in downstream unsolicited mode, so that the labels for the FEC are distributed other than along the SPT.

The above criteria are identical to those defined in [RFC5286].

In IP, a received FN message may result in rewriting the next-hop in the FIB. If LDP is applied, the label FIB also needs to be updated in accordance with the new next-hop; in the LFIB, however, not only the outgoing interface needs to be replaced but also the label that is

valid to this non-default next-hop. The latter is available due to liberal label retention and unsolicited downstream mode.

5.5. Application to VPN PE Protection

Protecting against (egress) PE router failures in VPN scenarios is conceptually similar to protecting against ASBR failures for Internet traffic. The difference is that in case of ASBR protection core routers are normally aware of external prefixes using iBGP, while in VPN cases P routers can only route inside the domain. In case of VPNs, tunnels running between ingress PE and egress PE decrease the burden for P routers. The task here is to redirect traffic to a backup egress PE.

Egress PE protection effectively calls out for an explicit failure notification, yet existing proposals try to avoid it.

[I-D.bashandy-bgp-edge-node-frr] proposes that the P routers adjacent to the primary PE maintain the necessary routing state and perform the tunnel decaps/re-encaps to the backup PE, thereby proposing considerable complexity for P routers.

[I-D.ietf-pwe3-redundancy] describes a mechanism for pseudowire redundancy, where PE routers need to run multi-hop BFD sessions to detect the loss of a primary egress PE. This leads to a potentially full mesh of multihop BFD session, which is a tremendous complexity. In addition, in some cases the egress PE of the secondary PW might need to explicitly set the PW state from standby to active.

FN provides the needed mechanism to actively inform all nodes including PE routers that a failure happened, and also identifies that a node failure happened. Furthermore, since both the ingress PE and the secondary egress PE are informed, all information is available for a proper switch-over. This is without a full mesh of BFD sessions running all the time between PE routers.

5.6. Bypassing Legacy Nodes

Legacy nodes, while cannot originate fast notifications and cannot process them either, can be assumed to be able to forward the notifications. As [fn-transport] discusses, FN forwarding is based on multicast. It is safe to assume that legacy routers' multicast configuration can be set up statically so as to be able to propagate fast notifications as needed.

When calculating failure specific alternative routes, IPFRR-FN capable nodes must consider legacy nodes as being fixed directed

links since legacy nodes do not change packet forwarding in the case of failure. There are situations when an FN-IPFRR capable node can, exceptionally, bypass a non-IPFRR-FN capable node in order to handle a remote failure.

As an example consider the topology depicted in Figure 9, where the link between C and D fails. C cannot locally repair the failure.

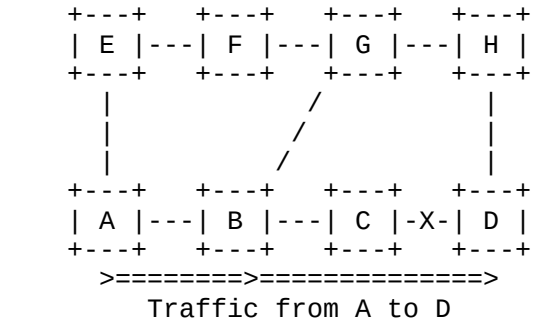


Figure 9 Example for bypassing legacy nodes

First, let us assume that each node is IPFRR-FN capable. C would advertise the failure information using FN. Each node learns that the link between C and D fails, as a result of which C changes its forwarding table to send any traffic destined to D via B. B also makes a change, replacing its default next-hop (C) with G. Note that other nodes do not need to modify their forwarding at all.

Now, let us assume that B is a legacy router not supporting IPFRR-FN but it is statically configured to multicast fast notifications as needed. As such, A will receive the notification. A's pre-calculations have been done knowing that B is unable to correct the failure. Node A, therefore, has pre-calculated E as the failure specific next-hop. Traffic entering at A and heading to D can thus be repaired.

5.7. Capability Advertisement

The solution requires nodes to know which other nodes in the area are capable of IPFRR-FN. The most straightforward way to achieve this is to rely on the Router Capability TLVs available both in OSPF [RFC4970] and in IS-IS [RFC4971].

5.8. Constraining the Dissemination Scope of Fast Notification Packets

As discussed earlier in Section 4.4. it is desirable to constrain the dissemination scope of failure notification messages. This section presents three candidate methods for controlling the scope of failure notification: (1) Pre-configure the TTL for FN messages in routers based on best current practices and related studies of available ISP and enterprise network topologies; (2) dynamically calculate the minimum TTL value needed to ensure 100% remote LFAP coverage; and (3) dynamically calculate the set of neighbours for which FN message should given the identity of the link that has failed.

These candidate dissemination options are mechanisms with different levels of optimality and complexity. The intent here is to present some options that will generate further discussion on the tradeoffs between different FN message scoping methods.

5.8.1. Pre-Configured FN TTL Setting

As discussed, earlier in Section 4.4. studies of various network topologies suggest that a fixed TTL setting of 2 hops may be sufficient to ensure failure notification message for typical OSPF area topologies. Therefore, a potentially simple solution for constraining FN message dissemination is for network managers to configure their routers with fixed TTL setting (e.g., TTL=2 hops) for FN messages. This TTL setting can be adjusted by network managers to consider implementation-specific details of the topology such as configuring a larger TTL setting for topologies containing, say, large ring sub-graph structures.

In terms of performance trades, pre-configuring the FN TTL, since it is fixed at configuration time, incurs no computational overhead for the router. On the other hand, it represents a configurable router parameter that network administrators must manage. Furthermore, the fixed, pre-configured FN TTL approach is sub-optimal in terms of constraining the FN dissemination as most single link events will not require FN messages send to up to TTL hops away from the failure site.

5.8.2. Advanced FN Scoping

While the static pre-configured setting of the FN TTL will likely work in practice for a wide range of OSPF area topologies, it has at two least weaknesses: (1) There may be certain topologies for which the TTL setting happens to be insufficient to provide the needed failure coverage; and (2) as discussed above, it tends to result in

FN being disseminated to a larger radius than needed to facilitate re-routing.

The solution to these drawbacks is for routers to dynamically compute the FN TTL radius needed for each of the local links it monitors. Doing so addresses the two weakness of a pre-configured TTL setting by computing a custom TTL setting for each of its local links that matches exactly the FN message radius for the given topology. The drawback, of course, is the additional computations. However, given a quasi-static network topology, it is possible this dynamic FN TTL computation is performed infrequently and, therefore, on average incurs relatively small computation overhead.

While a pre-configured TTL eliminates computation overhead at the expense of FN dissemination overhead and dynamic updates of the TTL settings achieve better dissemination efficiency by incurring some computational complexity, directed FN message forwarding attempts to minimize the FN dissemination scope by leveraging additional computation power. Here, rather than computing a FN TTL setting for each local link, a network employing directed forwarding has each router instance R compute the sets of one-hop neighbours to which a FN message must be forwarded for every possible failure event in the routing area. This has the beneficial effect of constraining the FN scope to the direction where there are nodes that require the FN update as opposed to disseminating to the entire TTL hop radius about a failure site. The trade off here, of course, is the additional computation complexity incurred and the maintenance of forwarding state for each possible failure case. Reference [Cev2010] gives an algorithm for finding, for each failure event, the direct neighbours to which the notification should be forwarded.

6. Protection against Replay Attacks

To defend against replay attacks, recipients should be able to ignore a re-sent recording of a previously sent FN packet. This suggests that some sort of sequence number should be included in the FN packet, the verification of which should not need control plane involvement. Since the solution should be simple to implement in the dataplane, maintaining and verifying per-source sequence numbers is not the best option.

We propose, therefore, that messages should be stamped with the digest of the actual routing configuration, i.e., a digest of the link state database of the link state routing protocol. The digest has to be picked carefully, so that if two LSDBs describe the same connectivity information, their digest should be identical as well,

and different LSDBs should result in different digest values with high probability.

The conceptual way of handling these digests could be the following:

- When the LSDB changes, the IGP re-calculates the digest and downloads the new value to the dataplane element(s), in a secure way.
- When a FN packet is originated, the digest is put into the FN message into the Sequence Number field.
- Network nodes distribute (forward) the FN packet.
- When processing, the dataplane element first performs an authentication check of the FN packet, as described in [fn-transport].
- Finally, before processing the failure notification, the dataplane element should check whether its own known LSDB digest is identical with the one in the message.

If due to a failure event a node disseminates a failure notification with FN, an attacker might capture the whole packet and re-send it later. If it resends the packet after the IGP re-converged on the new topology, the active LSDB digest is different, so the packet can be ignored. If the packet is replayed to a recipient who still has the same LSDB digest, then it means that the original failure notification was already processed but the IGP has not yet finished converging; the IPFRR detour is already active, the replica has no impact.

6.1. Calculating LSDB Digest

We propose to create an LSDB digest that is conceptually similar to [ISISDigest]. The operation is proposed to be the following:

- Create a hash from each LSA(OSPF)/LSP(ISIS) one by one
- XOR these hashes together
- When an LSA/LSP is removed, the new LSDB digest is received by computing the hash of the removed LSA, and then XOR to the existing digest

- When an LSA/LSP is added, the new LSDB digest is received by computing the hash of the new LSA, and then XOR to the existing digest

7. Security Considerations

The IPFRR application of Fast Notification does not raise further known security consideration in addition to those already present in Fast Notification itself. If an attacker could send false Failure Identification Messages or could hinder the transmission of legal messages, then the network would produce an undesired routing behaviour. These issues should be solved, however, in [fn-transport].

IPFRR-FN relies on the authentication mechanism provided by the Fast Notification transport protocol [fn-transport]. The specification of the FN transport protocol requires applications to protect against replay attacks with application specific sequence numbers. This draft, therefore, describes its own proposed sequence number in Section 5.8.1.

8. IANA Considerations

The Failure Identification message types need to be allocated a value in the FN App Type field.

IPFRR-FN capability needs to be allocated within Router Capability TLVs both for OSPF [RFC4970] and in IS-IS [RFC4971].

9. References

9.1. Normative References

[RFC5286]
A. Atlas, A. Zinin, "Basic specification for IP Fast-Reroute: Loop-Free Alternates", Internet Engineering Task Force: RFC 5286, 2008.

[fn-transport]
W. Lu, S. Kini, A. Csaszar, G. Enyedi, J. Tantsura, A. Tian, "Transport of Fast Notifications Messages", draft-lu-fn-transport, 2011

[RFC4970]
A. Lindem et al., Extensions to OSPF for Advertising Optional Router Capabilities, RFC 4970, 2007

[RFC4971]
JP. Vasseur et al., Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information, RFC 4971, 2007

[RFC4203]
K. Kompella, Y. Rekhter, " OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC4203, 2005

9.2. Informative References

[BFD]
D. Katz, D. Ward, "Bidirectional forwarding detection", RFC 5880, IETF, 2010

[RFC5714]
M. Shand, S. Bryant, "IP Fast Reroute Framework", RFC 5714, IETF, 2010.

[Cev2010]
S. Sevher, T. Chen, I. Hokelek, J. Kang, V. Kaul, Y.J. Lin, M. Pang, R. Rodoper, S. Samtani, C. Shah, J. Bowcock, G. B. Rucker, J. L. Simbol and A. Staikos, "An Integrated Soft Handoff Approach in IP Fast Reroute in Wireless Mobile Networks", In Proceedings IEEE COMSNETS, 2011.

[Eny2009]
Gabor Enyedi, Peter Szilagyi, Gabor Retvari, Andras Csaszar, "IP Fast ReRoute: Lightweight Not-Via without Additional Addresses", IEEE INFOCOM-MiniConference, Rio de Janeiro, Brazil, 2009.

[FIFR]
J. Wand, S. Nelakuditi, "IP fast reroute with failure inferencing", In Proceedings of ACM SIGCOMM Workshop on Internet Network Management - The Five-Nines Workshop, 2007.

[Hok2007]
I. Hokelek, M. A. Fecko, P. Gurung, S. Samtani, J. Sucec, A. Staikos, J. Bowcock and Z. Zhang, "Seamless Soft Handoff in Wireless Battlefield Networks Using Local and Remote LFAPs", In Proceedings IEEE MILCOM, 2007.

[Hok2008]
I. Hokelek, S. Cevher, M. A. Fecko, P. Gurung, S. Samtani, Z. Zhang, A. Staikos and J. Bowcock, "Testbed Implementation of Loop-Free Soft Handoff in Wireless Battlefield Networks", In Proceedings of the 26th Army Science Conference, December 1-4, 2008.

[MRC]
T. Cicic, A. F. Hansen, A. Kvalbein, M. Hartmann, R. Martin, M. Menth, S. Gjessing, O. Lysne, "Relaxed multiple routing configurations IP fast reroute for single and correlated failures", IEEE Transactions on Network and Service Management, available online: http://www3.informatik.uni-wuerzburg.de/staff/menth/Publications/papers/Menth08-Sub-4.pdf, September 2010.

[NotVia]
S. Bryant, M. Shand, S. Previdi, "IP fast reroute using Not-via addresses", Internet Draft, draft-ietf-rtgwg-ipfrr-notvia-addresses, 2010.

[RLFAP]
I. Hokelek, M. Fecko, P. Gurung, S. Samtani, S. Cevher, J. Sucec, "Loop-Free IP Fast Reroute Using Local and Remote LFAPs", Internet Draft, draft-hokelek-rlfap-01 (expired), 2008.

[Ret2011]
G. Retvari, J. Tapolcai, G. Enyedi, A. Csaszar, "IP Fast ReRoute: Loop Free Alternates Revisited", to appear at IEEE INFOCOM 2011

[ISISDigest]
J. Chiabaut and D. Fedyk. IS-IS Multicast Synchronization Digest. Available online: http://www.ieee802.org/1/files/public/docs2008/aq-fedyk-ISIS-digest-1108-v1.pdf, Nov 2008.

[BGPAddPaths]
D. Walton, A. Retana, E. Chen, J. Scudder, "Advertisement of Multiple Paths in BGP", draft-ietf-idr-add-paths, Work in progress

[DiverseBGP]
R. Raszuk, et. Al, "Distribution of diverse BGP paths", draft-ietf-grow-diverse-bgp-path-dist, Work in progress

[BGPBestExt]
P. Marques, R. Fernando, E. Chen, P. Mohapatra, H. Gredler, "Advertisement of the best external route in BGP", draft-ietf-idr-best-external, Work in progress

[BRITE]
Oliver Heckmann et al., "How to use topology generators to create realistic topologies", Technical Report, Dec 2002.

[MRT-ARCH]
A. Atlas et al., "An Architecture for IP/LDP Fast-Reroute Using Maximally Redundant Trees", Internet Draft, draft-ietf-rtgwg-mrt-frr-architecture-01, 2012

[MRT-ALG]
A. Atlas, G. Enyedi, A. Csaszar, "Algorithms for computing Maximally Redundant Trees for IP/LDP Fast-Reroute", Internet Draft, draft-enyedi-rtgwg-mrt-frr-algorithm-01, 2012

[I-D.ietf-pwe3-redundancy]
P. Muley, M. Aissaoui, M. Bocci, "Pseudowire Redundancy", draft-ietf-pwe3-redundancy (Work in progress!), May 2012

[I-D.bashandy-bgp-edge-node-frr]
A. Bashandy, B. Pithawala, K. Patel, "Scalable BGP FRR Protection against Edge Node Failure", draft-bashandy-bgp-edge-node-frr (Work in progress!), March 2012

10. Acknowledgments

The authors would like to thank Albert Tian, Wenhu Lu, Acee Lindem and Ibrahim Hokelek for the continuous discussions and comments on the topic, as well as Joel Halpern for his comments and review.

Appendix A. Memory Needs of a Naive Implementation

Practical background might suggest that storing and maintaining backup next-hops for many potential remote failures could overwhelm the resources of router linecards. This section attempts to provide a calculation describing the approximate memory needs in reasonable sized networks with a possible implementation.

A.1. An Example Implementation

Let us suppose that for exterior destinations the forwarding engine is using recursive lookup or indirection in order to improve updating time such as described in Section 5.3. We are also supposing that the concept of "prefix groups" is applied, i.e. there is an internal entity for the prefixes using exactly the same primary and backup ASBRs, and the next hop entry for a prefix among them is pointing to the next hop towards this entity. See e.g. Figure 7.

In the sequel, the term of "area" refers to an extended area, made up by the OSPF or IS-IS area containing the router, with the prefix groups added to the area as virtual nodes. Naturally, a prefix group is connected to the egress routers (ABRs) through which it can be reached. We just need to react to the failure ID of an ASBR for all the prefix groups connected to that ASBR; technically, we must suppose that one of the virtual links of all the affected prefix groups go down.

Here we show a simple naive implementation which can easily be beaten in real routers. This implementation uses an array for all the nodes (including real routers and virtual nodes representing prefix groups) in the area (node array in the sequel), made up by two pointers and a length filed (an integer) per record. One of the pointers points to another array (called alternative array). That second array is basically an enumeration containing the IDs of those failures influencing a shortest path towards that node and an alternative neighbor, which can be used, when such a failure occurs. When a failure is detected, (either locally, or by FN), we can easily find the proper record in all the lists. Moreover, since these arrays can be sorted based on the failure ID, we can even use binary search to find the needed record. The length of this array is stored in the record of the node array pointing to the alternative list.

Now, we only need to know, which records in the FIB should be updated. Therefore there is a second pointer in the node array pointing to that record.

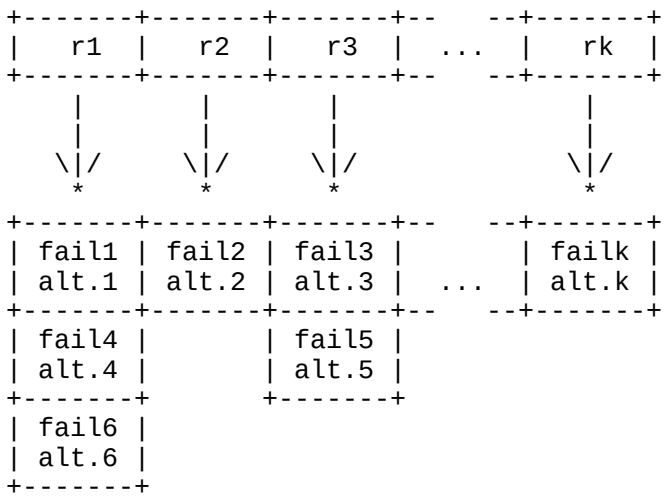


Figure 10The way of storing alternatives

A.2. Estimation of Memory Requirements.

Now, suppose that there are V nodes in the extended area, the network diameter is D, a neighbor descriptor takes X bytes, a failure ID takes Y bytes and a pointer takes Z bytes. We suppose that lookup for external prefixes are using indirection, so we only need to deal with destinations inside the extended area. In this way, if there is no ECMP, this data structure takes

(2*Z+Y)*(V-1) + 2*(X+Y)*D*(V-1)

bytes altogether. The first part is the memory consumption of the node array. The memory needed by alternative arrays: any path can contain at most D nodes and D links, each record needs X+Y bytes; there are records for all the other nodes in the area (V-1 nodes). Observe that this is a very rough overestimation, since most of the possible failures influencing the path will not change the next hop.

For computing memory consumption, suppose that neighbor descriptors, failure IDs and pointers take 4 bytes, there are 10000 nodes in the extended area (so both real routers and virtual nodes representing prefix groups are included) and the network diameter is 20 hops. In this case, we get that the node array needs about 120KB, the alternative array needs about 3.2MB, so altogether 3.4MB if there is no ECMP. Observe that the number of external prefixes is not important.

If however, there are paths with equal costs, the size of the alternative array increases. Suppose that there are 10 equal paths between ANY two nodes in the network. This would cause that the alternative list gets 10 times bigger, and now it needs a bit less than 32MB. Observe that the node array still needs only about 160KB, so 32MB is a good overestimation, which is likely acceptable for modern linecards with gigs of DRAM. Moreover, we need to stress here again that this is an extremely rough overestimation, so in reality much less memory will be enough. Furthermore, usually only protecting outer prefixes is needed, so we only need to protect the paths towards the prefix groups, which further decreases both the size of node array and the number of alternative lists.

A.3. Estimation of Failover Time

After a failover was detected either locally or by using FN, the nodes need to change the entries in their FIB. Here we do a rough estimation to show that the previous implementation can do it in at most a few milliseconds.

We are supposing that we have the data structure described in the previous section. When a failure happens we need to decide for each node in the node table whether the shortest path towards that destination was influenced by the failure. We can sort the elements in the alternative list, so now we can use binary search, which needs ceil(log(2D)) memory access (log here has base 2) for worst case. We need one more access to get the node list entry and another to rewrite the FIB.

We suppose DDR3 SDRAM with 64 byte cache line, which means that up to 8 entries of the alternative list can be fetched from the RAM at a time, so the previous formula is modified as we need ceil(log(D/4))+2 transactions. In this way for D=20 and V=10.000 we need (3+2)*10.000=50.000 transactions. If we suppose 10 ECMP paths as previously, D=200 and we need (5+2)*10000=70.000 transactions.

We can do a very conservative estimation by supposing a recent DDR3 SDRAM module which can do 5MT/s with completely random access, so doing 50.000 or 70.000 transaction takes 10ms or 14ms. Keep in mind that we assumed that there is only one memory controller, we always got the result of the search with the last read, and all the alternative lists were full. Moreover, internal system latencies (e.g. multiple memory requests) were overestimated seriously, since a DDR3 SDRAM can reach even 6 times this speed with random access.

Appendix B. Impact Scope of Fast Notification

The memory and fail-over time calculations presented in Appendix A are based on worst-case estimation. They assume that basically in a network with diameter equal to 20 hops, each failure has a route changing consequence on all routers in the full diameter.

This section provides experimental results on real-world topologies, showing that already 100% failure coverage can be achieved within a 2-hop radius around the failure.

We performed the coverage analysis of the fast reroute mechanism presented here on realistic topologies, which were generated by the BRITE topology generator in bottom-up mode [BRITE]. The coverage percentage is defined here as the percentage of the number of useable backup paths for protecting the primary paths which are failed because of link failures to the number of all failed primary paths.

The realistic topologies include AT&T and DFN using pre-determined BRITE parameter values from [BRITE] and various random topologies with different number of nodes and varying network connectivity. For example, the number of nodes for AT&T and DFN are 154 and 30, respectively, while the number of nodes for other random topologies is varied from 20 to 100. The BRITE parameters which are used in our topology generation process are summarized in Figure 11 (see [BRITE] for the details of each parameter). In summary, m represents the average number of edges per node and is set to either 2 or 3. A uniform bandwidth distribution in the range 100-1024 Mbps is selected and the link cost is obtained deterministically from the link bandwidth (i.e., inversely proportional to the link bandwidth as used by many vendors). Since the values for p(add) and beta determine the number of edges in the generated topologies, their values are varied to obtain network topologies with varying connectivity (e.g., sparse and dense).

	Bottom up
Grouping Model	Random pick
Model	GLP
Node Placement	Random
Growth Type	Incremental
Preferential Connectivity	On
BW Distribution	Uniform
Minimum BW	100
Maximum BW	1024
m	2-3
Number of Nodes (N)	20,30,50,100,154
p(add)	0.01,0.05,0.10,0.42
beta	0.01,0.05,0.15,0.62

Figure 11 BRITE topology generator parameters

The coverage percentage of our fast reroute method is reported for different network topologies (e.g., different number of nodes and varying network connectivity) using neighborhood depths of 0, 1, and 2. (i.e., X=0, 1, and 2). For a particular failure, backup routes protecting the failed primary paths are calculated only by those nodes which are within the selected radious of this failure. Note that these nodes are determined by the parameter X as follows: For X=0, two nodes which are directly connected to the failed link, for X=1, two nodes which are directly connected to the failed link and also neighboring nodes which are adjacent to one of the outgoing links of these two nodes, and so on.

The coverage percentage for a certain topology is computed by the following formula: Coverage Percentage = N_backupsexist*100/N_fpp where N_backupsexist is the number of source-destination pairs whose primary paths are failed because of link failures and have backup paths for protecting these failed paths, and N_fpp is the number of source-destination pairs whose primary paths are failed because of link failures. The source-destination pairs, in which source and destination nodes do not have any physical connectivity after a failure, are excluded from N_fpp. Note that the coverage percentage includes a network-wide result which is calculated by averaging all coverage results obtained by individually failing all edges for a certain network topology.

Figure 12 shows the coverage percentage results for random topologies with different number of nodes (N) and network connectivity, and Figure 13 shows these results for AT&T and DFN topologies. In these

figures, E_mean represents the average number of edges per node for a certain topology. Note that the average number of edges per node is determined by the parameters m, p(add), and beta. We observed that E_mean increases when p(add) and beta values increase. For each topology, coverage analysis is repeated for 10 topologies generated randomly by using the same BRITE parameters. E_mean and coverage percentage are obtained by averaging the results of these ten experiments.

Case	N	E_mean	X=0	X=1	X=2
p(add)=0.01 beta=0.01	20	3.64	82.39	98.85	100.0
	50	3.86	82.10	98.69	100.0
	100	3.98	83.21	98.04	100.0
p(add)=0.05 beta=0.05	20	3.70	85.60	99.14	100.0
	50	4.01	84.17	99.09	100.0
	100	4.08	83.35	98.01	100.0
p(add)=0.1 beta=0.15	20	5.52	93.24	100.0	100.0
	50	6.21	91.46	99.87	100.0
	100	6.39	91.17	99.86	100.0

Figure 12 Coverage percentage results for random topologies

Case	N	E_mean	X=0	X=1	X=2
p(add)=0.42 beta=0.62	154 (AT&T)	6.88	91.04	99.81	100.0
	30 (DFN)	8.32	93.76	100.0	100.0

Figure 13 Coverage percentage results for AT&T and DFN topologies

There are two main observations from these results:

1. As the neighborhood depth (X) increases the coverage percentage increases and the complete coverage is obtained using a low neighborhood depth value (i.e., X=2). This result is significant since failure notification message needs to be sent only to nodes which are two-hop away from the point of failure for the complete

coverage. This result supports that our method provides fast convergence by introducing minimal signaling overhead within only the two-hop neighborhood.

2. The topologies with higher connectivity (i.e., higher E_mean values) have better coverage compared to the topologies with lower connectivity (i.e., lower E_mean values). This is an intuitive result since the number of possible alternate hops in dense network topologies is higher than the number of possible alternate hops in sparse topologies. This phenomenon increases the likelihood of finding backup paths, and therefore the coverage percentage.

Authors' Addresses

Andras Csaszar
Ericsson
Irinyi J utca 4-10, Budapest, Hungary, 1117
Email: Andras.Csaszar@ericsson.com

Gabor Sandor Enyedi
Ericsson
Irinyi J utca 4-10, Budapest, Hungary, 1117
Email: Gabor.Sandor.Enyedi@ericsson.com

Jeff Tantsura
Ericsson
300 Holger Way, San Jose, CA 95134
Email: jeff.tantsura@ericsson.com

Sriganesh Kini
Ericsson
300 Holger Way, San Jose, CA 95134
Email: sriganesh.kini@ericsson.com

John Sucec
Telcordia Technologies
One Telcordia Drive, Piscataway, NJ 08854
Email: sucecj@telcordia.com

Subir Das
Telcordia Technologies
One Telcordia Drive, Piscataway, NJ 08854
Email: sdas2@telcordia.com

Network Working Group
Internet Draft
Intended status: Proposed Standard
Expires: March 2012

S. Giacalone
Thomson Reuters

D. Ward
Juniper Networks

J. Drake
Juniper Networks

A. Atlas
Juniper Networks

S. Previdi
Cisco Systems

September 21, 2011

OSPF Traffic Engineering (TE) Express Path
draft-giacalone-ospf-te-express-path-02.txt

Abstract

In certain networks, such as, but not limited to, financial information networks (e.g. stock market data providers), network performance criteria (e.g. latency) are becoming as critical to data path selection as other metrics.

This document describes extensions to OSPF TE [RFC3630] such that network performance information can be distributed and collected in a scalable fashion. The information distributed using OSPF TE Express Path can then be used to make path selection decisions based on network performance.

Note that this document only covers the mechanisms with which network performance information is distributed. The mechanisms for measuring network performance or acting on that information, once distributed, are outside the scope of this document.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Copyright Notice

Table of Contents

1. Introduction

In certain networks, such as, but not limited to, financial information networks (e.g. stock market data providers), network performance information (e.g. latency) is becoming as critical to data path selection as other metrics.

In these networks, extremely large amounts of money rest on the ability to access market data in "real time" and to predictably make trades faster than the competition. Because of this, using metrics such as hop count or cost as routing metrics is becoming only

tangentially important. Rather, it would be beneficial to be able to make path selection decisions based on performance data (such as latency) in a cost-effective and scalable way.

This document describes extensions to OSPF TE (hereafter called "OSPF TE Express Path"), that can be used to distribute network performance information (such as link delay, delay variation, packet loss, residual bandwidth, and available bandwidth).

The data distributed by OSPF TE OSPF TE Express Path is meant to be used as part of the operation of the routing protocol (e.g. by replacing cost with latency or considering bandwidth as well as cost), by enhancing CSPF, or for other uses such as supplementing the data used by an Alto server [Alto]. With respect to CSPF, the data distributed by OSPF TE Express Path can be used to setup, fail over, and fail back data paths using protocols such as RSVP-TE [RFC3209].

Note that the mechanisms described in this document only disseminate performance information. The methods for initially gathering that performance information, such as [Frost], or acting on it once it is distributed are outside the scope of this document.

2. Conventions used in this document

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

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

3. Express Path Extensions to OSPF TE

This document proposes new OSPF TE sub-TLVs that can be announced in OSPF TE LSAs to distribute network performance information. The extensions in this document build on the ones provided in OSPF TE [RFC3630] and GMPLS [RFC4203].

OSPF TE LSAs [RFC3630] are opaque LSAs [RFC5250] with area flooding scope. Each TLV has one or more nested sub-TLVs which permit the TE LSA to be readily extended. There are two main types of OSPF TE LSA; the Router Address or Link TE LSA. Like the extensions in GMPLS

(RFC4203), this document proposes several additional sub-TLVs for the Link TE LSA:

Type	Length	Value
TBD1	4	Unidirectional Link Delay
TBD2	4	Unidirectional Delay Variation
TBD3	4	Unidirectional Packet Loss
TBD4	4	Unidirectional Residual Bandwidth Sub TLV
TBD5	4	Unidirectional Available Bandwidth Sub TLV

As can be seen in the list above, the sub-TLVs described in this document carry different types of network performance information. Many (but not all) of the sub-TLVs include a bit called the Anomalous (or "A") bit. When the A bit is clear (or when the sub-TLV does not include an A bit), the sub-TLV describes steady state link performance. This information could conceivably be used to construct a steady state performance topology for initial tunnel path computation, or to verify alternative failover paths.

When network performance violates configurable link-local thresholds a sub-TLV with the A bit set is advertised. These sub-TLVs could be used by the receiving node to determine whether to fail traffic to a backup path, or whether to calculate an entirely new path. From an MPLS perspective, the intent of the A bit is to permit LSP ingress nodes to:

A) Determine whether the link referenced in the sub-TLV affects any of the LSPs for which it is ingress. If there are, then:

B) Determine whether those LSPs still meet end-to-end performance objectives. If not, then:

C) The node could then conceivably move affected traffic to a pre-established protection LSP or establish a new LSP and place the traffic in it.

If link performance then improves beyond a configurable minimum value (reuse threshold), that sub-TLV can be re-advertised with the Anomalous bit cleared. In this case, a receiving node can conceivably do whatever re-optimization (or failback) it wishes to do (including nothing).

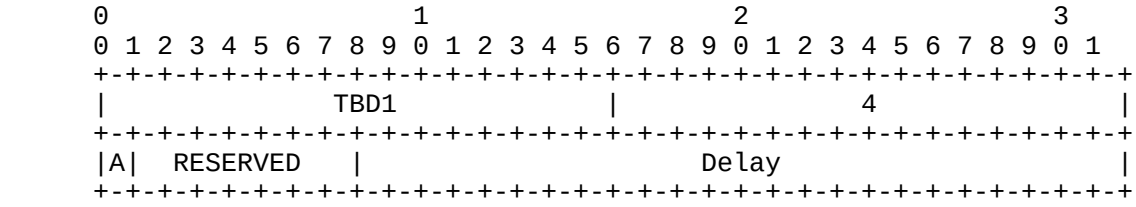
Note that when a sub-TLV does not include the A bit, that sub-TLV cannot be used for failover purposes. The A bit was intentionally omitted from some sub-TLVs to help mitigate oscillations. See section 7. 1. for more information.

Consistent with existing OSPF TE specifications (RFC3630), the bandwidth advertisements defined in this draft MUST be encoded as IEEE floating point values. The delay and delay variation advertisements defined in this draft MUST be encoded as integer values. Delay values MUST be quantified in units of microseconds, packet loss MUST be quantified as a percentage of packets sent, and bandwidth MUST be sent as bytes per second. All values (except residual bandwidth) MUST be calculated as rolling averages where the averaging period MUST be a configurable period of time. See section 5. for more information.

4. Sub TLV Details

4.1. Unidirectional Link Delay Sub-TLV

This sub-TLV advertises the average link delay between two directly connected OSPF neighbors. The delay advertised by this sub-TLV MUST be the delay from the local neighbor to the remote one (i.e. the forward path latency). The format of this sub-TLV is shown in the following diagram:



4.1.1. Type

This sub-TLV has a type of TBD1.

4.1.2. Length

The length is 4.

4.1.3. A bit

This field represents the Anomalous (A) bit. The A bit is set when the measured value of this parameter exceeds its configured maximum threshold. The A bit is cleared when the measured value falls below its configured reuse threshold. If the A bit is clear, the sub-TLV represents steady state link performance.

4.1.4. Reserved

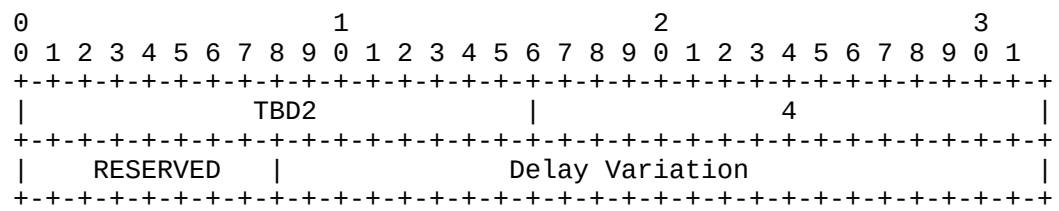
This field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

4.1.5. Delay Value

This 24-bit field carries the average link delay over a configurable interval in micro-seconds, encoded as an integer value. When set to 0, it has not been measured. When set to the maximum value 16,777,215 (16.777215 sec), then the delay is at least that value and may be larger.

4.2. Unidirectional Delay Variation Sub-TLV

This sub-TLV advertises the average link delay variation between two directly connected OSPF neighbors. The delay variation advertised by this sub-TLV MUST be the delay from the local neighbor to the remote one (i.e. the forward path latency). The format of this sub-TLV is shown in the following diagram:



4.2.1. Type

This sub-TLV has a type of TBD2.

4.2.2. Length

The length is 4.

4.2.3. Reserved

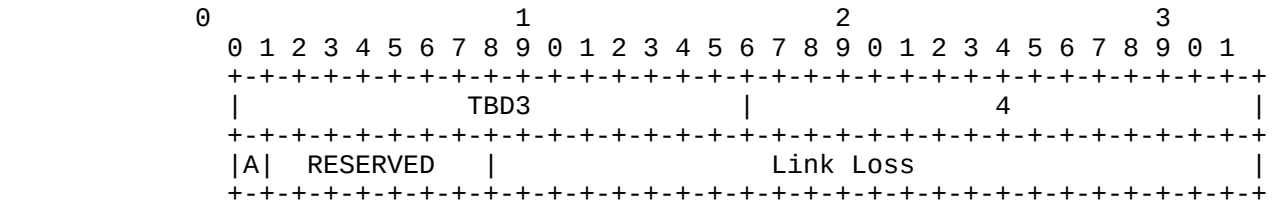
This field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

4.2.4. Delay Variation

This 24-bit field carries the average link delay variation over a configurable interval in micro-seconds, encoded as an integer value. When set to 0, it has not been measured. When set to the maximum value 16,777,215 (16.777215 sec), then the delay is at least that value and may be larger.

4.3. Unidirectional Link Loss Sub-TLV

This sub-TLV advertises the loss (as a packet percentage) between two directly connected OSPF neighbors. The link loss advertised by this sub-TLV MUST be the packet loss from the local neighbor to the remote one (i.e. the forward path loss). The format of this sub-TLV is shown in the following diagram:



4.3.1. Type

This sub-TLV has a type of TBD3

4.3.2. Length

The length is 4

4.3.3. A bit

This field represents the Anomalous (A) bit. The A bit is set when the measured value of this parameter exceeds its configured maximum threshold. The A bit is cleared when the measured value falls below

its configured reuse threshold. If the A bit is clear, the sub-TLV represents steady state link performance.

4.3.4. Reserved

This field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

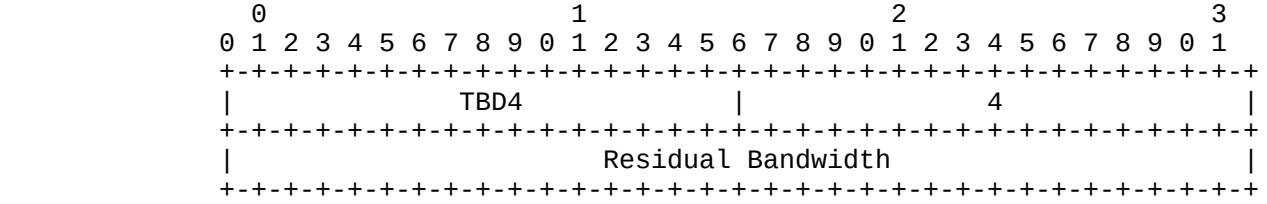
4.3.5. Link Loss

This 24-bit field carries link packet loss as a percentage of the total traffic sent over a configurable interval. The basic unit is 0.000003%, where (2^24 - 2) is 50.331642%. This value is the highest packet loss percentage that can be expressed (the assumption being that precision is more important on high speed links than the ability to advertise loss rates greater than this, and that high speed links with over 50% loss are unusable). Therefore, measured values that are larger than the field maximum SHOULD be encoded as the maximum value. When set to a value of all 1s (2^24 - 1), the link packet loss has not been measured.

4.4. Unidirectional Residual Bandwidth Sub-TLV

This TLV advertises the residual bandwidth (defined in section 4.4.3. between two directly connected OSPF neighbors. The residual bandwidth advertised by this sub-TLV MUST be the residual bandwidth from the system originating the LSA to its neighbor.

The format of this sub-TLV is shown in the following diagram:



4.4.1. Type

This sub-TLV has a type of TBD4.

4.4.2. Length

The length is 4.

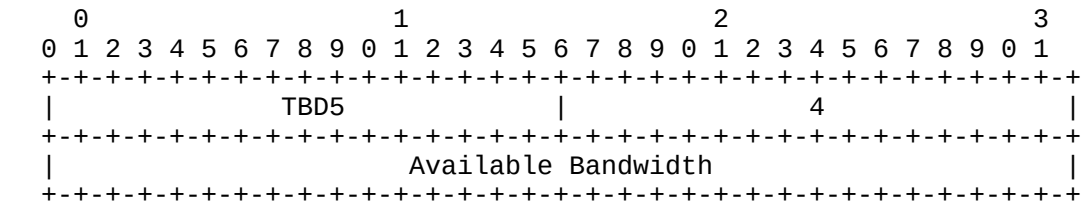
4.4.3. Residual Bandwidth

This field carries the residual bandwidth on a link, forwarding adjacency [RFC4206], or bundled link in IEEE floating point format with units of bytes per second. For a link or forwarding adjacency, residual bandwidth is defined to be Maximum Bandwidth [RFC3630] minus the bandwidth currently allocated to RSVP-TE LSPs. For a bundled link, residual bandwidth is defined to be the sum of the component link residual bandwidths.

Note that although it may seem possible to calculate Residual Bandwidth using the existing sub-TLVs in RFC 3630, this is not a consistently reliable approach and hence the Residual Bandwidth sub-TLV has been added here. For example, because the Maximum Reservable Bandwidth [RFC3630] can be larger than the capacity of the link, using it as part of an algorithm to determine the value of the Maximum Bandwidth [RFC3630] minus the bandwidth currently allocated to RSVP-TE LSPs cannot be considered reliably accurate.

4.5. Unidirectional Available Bandwidth Sub-TLV

This TLV advertises the available bandwidth (defined in section 4.4.6.) between two directly connected OSPF neighbors. The available bandwidth advertised by this sub-TLV MUST be the available bandwidth from the system originating the LSA to its neighbor. The format of this sub-TLV is shown in the following diagram:



4.4.4. Type

This sub-TLV has a type of TBD5.

4.4.5. Length

The length is 4.

4.4.6. Available Bandwidth

This field carries the available bandwidth on a link, forwarding adjacency, or bundled link in IEEE floating point format with units of bytes per second. For a link or forwarding adjacency, available bandwidth is defined to be residual bandwidth (see section 4.4.) minus the measured bandwidth used for the actual forwarding of non-RSVP-TE LSP packets. For a bundled link, available bandwidth is defined to be the sum of the component link available bandwidths.

5. Announcement Thresholds and Filters

The values advertised in all sub-TLVs MUST be controlled using an exponential filter (i.e. a rolling average) with a configurable measurement interval and filter coefficient.

Implementations are expected to provide separately configurable advertisement thresholds. All thresholds MUST be configurable on a per sub-TLV basis.

The announcement of all sub-TLVs that do not include the A bit SHOULD be controlled by variation thresholds that govern when they are sent.

Sub-TLV that include the A bit are governed by several thresholds. Firstly, a threshold SHOULD be implemented to govern the announcement of sub-TLVs that advertise a change in performance, but not an SLA violation (i.e. when the A bit is not set). Secondly, implementations MUST provide configurable thresholds that govern the announcement of sub-TLVs with the A bit set (for the indication of a performance violation). Thirdly, implementations SHOULD provide reuse thresholds. These thresholds govern sub-TLV re-announcement with the A bit cleared to permit fail back.

6. Announcement Suppression

When link performance average values change, but fall under the threshold that would cause the announcement of a sub-TLV with the A bit set, implementations MAY suppress or throttle sub-TLV

announcements. All suppression features and thresholds SHOULD be configurable.

7. Network Stability and Announcement Periodicity

To mitigate concerns about stability, all values (except residual bandwidth) MUST be calculated as rolling averages where the averaging period MUST be a configurable period of time, rather than instantaneous measurements.

Announcements MUST also be able to be throttled using configurable inter-update throttle timers. The minimum announcement periodicity is 1 announcement per second.

8. Compatibility

As per (RFC3630), unrecognized TLVs should be silently ignored

9. Security Considerations

This document does not introduce security issues beyond those discussed in [RFC3630] and [RFC5329].

10. IANA Considerations

IANA maintains the registry for the sub-TLVs. OSPF TE Express Path will require one new type code per sub-TLV defined in this document.

11. References

11.1. Normative References

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

[RFC3630] Katz, D., Kompella, K., Yeung, D., "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

11.2. Informative References

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

[RFC3031] Rosen, E., Viswanathan, A., Callon, R., "Multiprotocol Label Switching Architecture", 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.

[RFC5250] Berger, L., Bryskin I., Zinin, A., Coltun, R., "The OSPF Opaque LSA Option", RFC 5250, July 2008.

[Frost] D. Frost, S. Bryant"A Packet Loss and Delay Measurement Profile for MPLS-based Transport Networks"

[Alto] R. Alimi R. Penno Y. Yang, "ALTO Protocol"

12. Acknowledgments

The authors would like to recognize Ayman Soliman for his contributions.

This document was prepared using 2-Word-v2.0.template.dot.

13. Author's Addresses

Spencer Giacalone
Thomson Reuters
195 Broadway
New York NY 10007, USA

Email: Spencer.giacalone@thomsonreuters.com

Dave Ward
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089, USA

Email: dward@juniper.net

John Drake
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089, USA

Email: jdrake@juniper.net

Alia Atlas
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089, USA

Email: akatlas@juniper.net

Stefano Previdi
Cisco Systems
Via Del Serafico 200
00142 Rome
Italy

Email: sprevidi@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 21, 2012

Clarence Filsfils
Cisco Systems
Pierre Francois
Institute IMDEA Networks
January 18, 2012

LFA applicability in SP networks
draft-ietf-rtgwg-lfa-applicability-06

Abstract

In this document, we analyze the applicability of the Loop-Free Alternates method of providing IP fast re-route in both the core and the access parts of Service Provider networks. We consider both the link and node failure cases, and provide guidance on the applicability of LFA to different network topologies, with special emphasis on the access parts of the network.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 21, 2012.

Copyright Notice

Table of Contents

1. Introduction

In this document, we analyze the applicability of the Loop-Free Alternates (LFA) [RFC5714] [RFC5286] method of providing IP fast re-route (IPFRR) in both the core and the access parts of Service Provider (SP) networks. We consider both the link and node failure cases, and provide guidance on the applicability of LFA to different network topologies, with special emphasis on the access parts of the network.

We first introduce the terminology used in this document in Section 2. In Section 3, we describe typical access network designs and we analyze them for LFA applicability. In Section 4, we describe a simulation framework for the study of LFA applicability in SP core networks, and present results based on various SP networks. We then emphasize the independence between protection schemes used in the core and at the access level of the network. Finally we discuss the key benefits of LFA which stem from its simplicity and we draw some conclusions.

2. Terminology

We use IS-IS [RFC1195] as reference. It is assumed that normal routing (i.e., when traffic not being fast re-routed around a failure) occurs along the shortest path. The analysis is equally applicable to OSPF [RFC2328] [RFC5340].

A per-prefix LFA for a destination D at a node S is a precomputed backup IGP nexthop for that destination. This backup IGP nexthop can be link protecting or node protecting. In this document, we assume that all links to be protected with LFAs are point-to-point.

Link-protecting: A neighbor N is a link-protecting per-prefix LFA for S’s route to D if equation eq1 is satisfied, with eq1 == ND < NS + SD where XY refers to the IGP distance from X to Y. This is in line with the definition of an LFA in [RFC5714].

$$eq1 == ND < NS + SD$$

Equation eq1

Node-protecting: A Neighbor N is a node-protecting LFA for S’s route to D, with initial IGP nexthop F if N is a link-protecting LFA for D and equation eq2 is satisfied, with eq2 == ND < NF + FD. This is in line with the definition of a Node-Protecting Alternate Next-Hop in

[RFC5714].

eq2 == ND < NF + FD

Equation eq2

De facto node-protecting LFA: this is a link-protecting LFA that turns out to be node-protecting. This occurs in cases illustrated by the following examples :

- The LFA candidate that is picked by S actually satisfies Equation eq2 but S did not verify that property. The show command issued by the operator would not indicate this LFA as "node protecting" while in practice (de facto) it is.
- A cascading effect of multiple LFA's can also provide de facto node protection. Equation eq2 is not satisfied, but the combined activation of LFAs by some other neighbors of the failing node F provides (de facto) node protection. In other words, it puts the dataplane in a state such that packets forwarded by S ultimately reach a neighbor of F that has a node-protecting LFA. Note that in this case S cannot indicate the node-protecting behavior of the repair without running additional computations.

Per-Link LFA: a per-link LFA for the link SF is one precomputed backup IGP nexthop for all the destinations reached through SF. This is a neighbor of the repairing node that is a per-Prefix LFA for all the destinations that the repairing node reaches through SF. Note that such a per-link LFA exists if S has a per-prefix LFA for destination F.

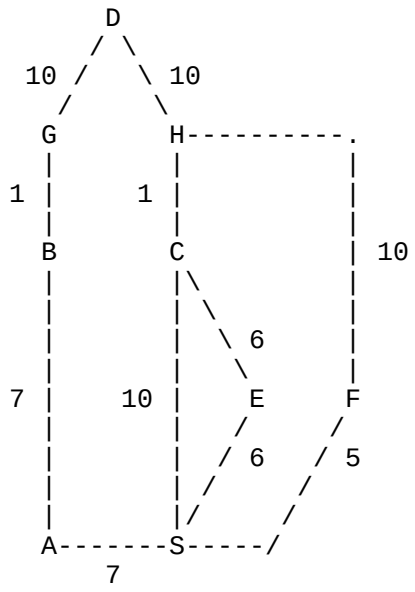


Figure 1: Example 1

In Figure 1, considering the protection of link SC, we can see that A, E, and F are per-prefix LFAs for destination D, as none of them use S to reach D.

For destination D, A and F are node-protecting LFA as they do not reach D through node C, while E is not node-protecting for S as it reaches D through C.

If S does not compute and select node-protecting LFAs, there is a chance that S picks the non node-protecting LFA E, although A and F were node-protecting LFAs. If S enforces the selection of node-protecting LFAs, then in the case of the single failure of link SC, S will first activate its LFA and deviate traffic addressed to D along S-A-B-G-D and/or S-F-H-D, and then converge to its post-convergence optimal path S-E-C-H-D.

A is not a per-link LFA for link SC because A reaches C via S. E is a per-Link LFA for link SC as it reaches C through link EC. This per-link LFA does not provide de facto node protection. Upon failure of node C, S would fast-reroute D-destined packets to its per-link lfa (= E). E would himself detect the failure of EC and hence activate its own per-link LFA (=S). Traffic addressed to D would be trapped in a loop and hence there is no de facto node protection behavior.

If there were a link between E and F, that E would pick as its LFA for destination D, then E would provide de facto node protection for S, as upon the activation of its LFA, S would deviate traffic addressed to D towards E, which in turns deviates that traffic to F, which does not reach D through C.

F is a per-Link LFA for link SC as F reaches C via H. This per-link LFA is de facto node-protecting for destination D as F reaches D via F-H-D.

MicroLoop (uLoop): the occurrence of a transient forwarding loop during a routing transition (as defined in [RFC5714]).

In Figure 1, the loss of link SE cannot create any uLoop because: 1/The link is only used to reach destination E and 2/ S is the sole node changing its path to E upon link SE failure. 3/ S's shortest path to E after the failure goes via C. 4/C's best path to E (before and after link SC failure) is via CE.

To the contrary, upon failure of link AB, a microloop may form for traffic destined to B. Indeed, if A updates its FIB before S, A will deviate B-destined traffic towards S, while S is still forwarding this traffic to A.

3. Access Network

The access part of the network often represents the majority of the nodes and links. It is organized in several tens or more of regions interconnected by the core network. Very often the core acts as an IS-IS level2 domain (OSPF area 0) while each access region is confined in an IS-IS level1 domain (OSPF non 0 area). Very often, the network topology within each access region is derived from a unique template common across the whole access network. Within an access region itself, the network is made of several aggregation regions, each following the same interconnection topologies.

For these reasons, in the next sections, we base the analysis of the LFA applicability in a single access region, with the following assumptions:

- Two routers (C1 and C2) provide connectivity between the access region and the rest of the network. If a link connects these two routers in the region area, then it has a symmetric IGP metric c.
- We analyze a single aggregation region within the access region. Two aggregation routers (A1 and A2) interconnect the aggregation region to the two routers C1 and C2 for the analyzed access region. If a link connects A1 to A2 then it has a symmetric IGP metric a. If a link connects an A to a C router then, for sake of

generality, we will call d the metric for the directed link CA and u the metric for the AC directed link.

- We analyze two edge routers E1 and E2 in the access region. Each is either dual-homed directly into C1 and C2 (Section 3.1) or into A1 and A2 (Section 3.2, Section 3.3, Section 3.4). The directed link metric between Cx/Ax and Ey is d and u in the opposite direction.
- We assume a multi-level IGP domain. The analyzed access region forms a level-1 (L1) domain. The core is the level-2 (L2) domain. We assume that the link between C1 and C2, if it exists, is configured as L1L2. We assume that the loopbacks of the C routers are part of the L2 topology. L1 routers learn about them as propagated routes (L2=>L1 with Down bit set). We remind that if an L1L2 router learns about X/x as an L1 path P1, an L2 path P2 and an L1L2 path P12, then it will prefer path P1. If P1 is lost, then it will prefer path P2.
- We assume that all the C, A and E routers may be connected to customers and hence we analyze LFA coverage for the loopbacks of each type of node.
- We assume that no useful traffic is directed to router-to-router subnets and hence we do not analyze LFA applicability for these.
- A prefix P models an important IGP destination that is not present in the local access region. The igp metric from C1 to P is x and the metric from C2 to P is x+e.
- We analyze LFA coverage against all link and node failures within the access region.
- WxYz refers to the link from Wx to Yz.
- We assume that c < d + u and a < d + u (commonly agreed design rule).
- In the square access design (Section 3.3), we assume that c < a (commonly agreed design rule).
- We analyze the most frequent topologies found in an access region.
- We first analyze per-prefix LFA applicability and then per-link.
- The topologies are symmetric with respect to a vertical axe and hence we only detail the logic for the link and node failures of the left half of the topology.

3.1. Triangle

We describe the LFA applicability for the failures of each direction of link C1E1, E1 and C1 (Figure 2), and for the failure of each node.

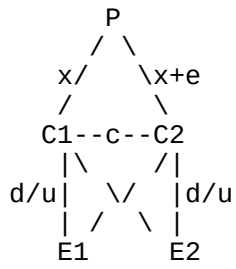


Figure 2: Triangle

3.1.1. E1C1 failure

3.1.1.1. Per-Prefix LFA

Three destinations are impacted by this link failure: C1, E2 and P.

The LFA for destination C1 is C2 because eq1 == c < d + u. Node protection for route C1 is not applicable. (if C1 goes down, traffic destined to C1 is lost anyway).

The LFA to E2 is via C2 because eq1 == d < d+u+d. It is node protecting because eq2 == d < c + d.

The LFA to P is via C2 because eq1 == c < d + u. It is node protecting if eq2 == x + e < x + c, i.e., if e < c. This relationship between e and c is an important aspect of the analysis, which is discussed in detail in Section 3.5 and Section 3.6

Conclusion: all important intra-PoP routes with primary interface E1C1 benefit from LFA link and node protection. All important inter-PoP routes with primary interface E1C1 benefit from LFA link protection, and also from node protection if e < c.

3.1.1.2. Per-Link LFA

We have a per-prefix LFA to C1 and hence we have a per-link LFA for link E1C1. All impacted destinations are protected for link failure. In case of C1 node failure, the traffic to C1 is lost (by definition), the traffic to E2 is de facto protected against node failure and the traffic to P is de facto protected when e < c.

3.1.2. C1E1 failure

3.1.2.1. Per-Prefix LFA

C1 has one single primary route via C1E1: the route to E1 (because c < d + u).

C1's LFA to E1 is via C2 because eq1 == d < c + d.

Node protection upon E1's failure is not applicable as the only impacted traffic is sinked at E1 and hence is lost anyway.

Conclusion: all important routes with primary interface C1E1 benefit from LFA link protection. Node protection is not applicable.

3.1.2.2. Per-Link LFA

We have a per-prefix LFA to E1 and hence we have a per-link LFA for link C1E1. De facto node protection is not applicable.

3.1.3. uLoop

The IGP convergence cannot create any uLoop. See Section 3.7.

3.1.4. Conclusion

All important intra-PoP routes benefit from LFA link and node protection or de facto node protection. All important inter-PoP routes benefit from LFA link protection. De facto node protection is ensured if e < c (this is particularly the case for dual-plane core or two-tiered-igp-metric design, see later sections).

The IGP convergence does not cause any uLoop.

Per-link LFA and per-Prefix LFA provide the same protection benefits.

3.2. Full-Mesh

We describe the LFA applicability for the failures of C1A1, A1E1, E1, A1 and C1 (Figure 3).

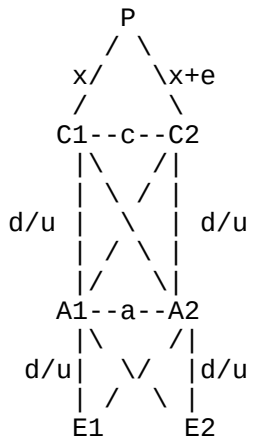


Figure 3: Full-Mesh

3.2.1. E1A1 failure

3.2.1.1. Per-Prefix LFA

Four destinations are impacted by this link failure: A1, C1, E2 and P.

The LFA for A1 is A2: eq1 == a < d + u. Node protection for route A1 is not applicable (if A1 goes down, traffic to A1 is lost anyway).

The LFA for C1 is A2: eq1 == u < d + u + u. Node protection for route C1 is guaranteed: eq2 == u < a + u.

The LFA to E2 is via A2: eq1 == d < d+u+d. Node protection is guaranteed: eq2 == d < a + d.

The LFA to P is via A2: eq1 == u + x < d + u + u + x. Node protection is guaranteed: eq2 == u+ x < a + u + x.

Conclusion: all important intra-PoP and inter-PoP routes with primary interface E1A1 benefit from LFA link and node protection.

3.2.1.2. Per-Link LFA

We have a per-prefix LFA to A1 and hence we have a per-link LFA for link E1A1. All impacted destinations are protected for link failure. De facto node protection is provided for all destinations (except to A1 which is not applicable).

3.2.2. A1E1 failure

3.2.2.1. Per-Prefix LFA

A1 has one single primary route via A1E1: the route to E1 (because c < d + u).

A1's LFA to E1 is via A2: eq1 == d < a + d.

Node protection upon E1's failure is not applicable as the only impacted traffic is sinked at E1 and hence is lost anyway.

Conclusion: all important routes with primary interface A1E1 benefit from LFA link protection. Node protection is not applicable.

3.2.2.2. Per-Link LFA

We have a per-prefix LFA to E1 and hence we have a per-link LFA for link C1E1. De facto node protection is not applicable.

3.2.3. A1C1 failure

3.2.3.1. Per-Prefix LFA

Two destinations are impacted by this link failure: C1 and P.

The LFA for C1 is C2 because eq1 == c < d + u. Node protection for route C1 is not applicable (if C1 goes down, traffic to C1 is lost anyway).

The LFA for P is via C2 because eq1 == c < d + u. It is de facto protected for node failure if eq2 == x + e < x + c.

Conclusion: all important intra-PoP routes with primary interface A1C1 benefit from LFA link protection (node protection is not applicable). All important inter-PoP routes with primary interface E1C1 benefit from LFA link protection (and from de facto node protection if e < c).

3.2.3.2. Per-Link LFA

We have a per-prefix LFA to C1 and hence we have a per-link LFA for link A1C1. All impacted destinations are protected for link failure. In case of C1 node failure, the traffic to C1 is lost (by definition) and the traffic to P is de facto node protected if e < c.

3.2.4. C1A1 failure

3.2.4.1. Per-Prefix LFA

C1 has three routes via C1A1: A1, E1 and E2. E2 behaves like E1 and hence is not analyzed further.

C1's LFA to A1 is via C2 because we assumed c < a and eq1 == d < c + d. Node protection upon A1's failure is not applicable as the traffic to A1 is lost anyway.

C1's LFA to E1 is via A2: eq1 == d < u+ d + d. Node protection upon A1's failure is guaranteed because: eq2 == d < a + d.

Conclusion: all important routes with primary interface C1A1 benefit from LFA link protection. Node protection is guaranteed where applicable.

3.2.4.2. Per-Link LFA

We have a per-prefix LFA to A1 and hence we have a per-link LFA for link C1E1. De facto node protection is available.

3.2.5. uLoop

The IGP convergence cannot create any uLoop. See Section 3.7.

3.2.6. Conclusion

All important intra-PoP routes benefit from LFA link and node protection.

All important inter-PoP routes benefit from LFA link protection. They benefit from node protection upon failure of A nodes. They benefit from node protections upon failure of C nodes if e < c (this is particularly the case for dual-plane core or two-tiered-igp-metric design, see later sections).

The IGP convergence does not cause any uLoop.

Per-link LFA and per-Prefix LFA provide the same protection benefits.

3.3. Square

We describe the LFA applicability for the failures of C1A1, A1E1, E1, A1 and C1 (Figure 4).

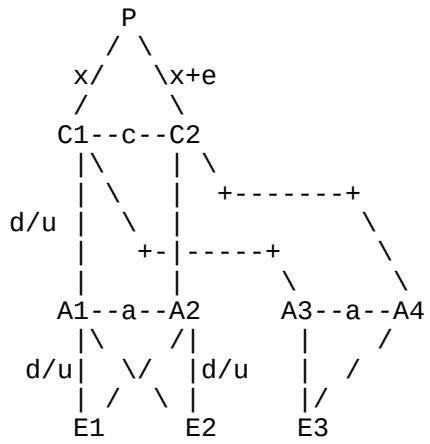


Figure 4: Square

3.3.1. E1A1 failure

3.3.1.1. Per-Prefix LFA

E1 has six routes via E1A1: A1, C1, P, E2, A3, E3.

E1's LFA route to A1 is via A2 because eq1 == a < d + u. Node protection for traffic to A1 upon A1 node failure is not applicable.

E1's LFA route to A3 is via A2 because eq1 == u + c + d < d + u + u + d. This LFA is guaranteed to be node protecting because eq2 == u + c + d < a + u + d.

E1's LFA route to C1 is via A2 because eq1 == u + c < d + u + u. This LFA is guaranteed to be node protecting because eq2 == u + c < a + u.

E1's primary route to E2 is via ECMP(E1A1, E1A2). The LFA for the first ECMP path (via A1) is the second ECMP path (via A2). This LFA is guaranteed to be node protecting because eq2 == d < a + d.

E1's primary route to E3 is via ECMP(E1A1, E1A2). The LFA for the first ECMP path (via A1) is the second ECMP path (via A2). This LFA is guaranteed to be node protecting because eq2 == u + d + d < a + u d + d.

If e=0: E1's primary route to P is via ECMP(E1A1, E1A2). The LFA for the first ECMP path (via A1) is the second ECMP path (via A2). This LFA is guaranteed to be node protecting because eq2 == u + x + 0 < a + u + x .

If e<>0: E1's primary route to P is via E1A1. Its LFA is via A2 because eq1 == u + c + x < d + u + u + x. This LFA is guaranteed to be node protecting because eq2 == u + c + x < a + u + x.

Conclusion: all important intra-PoP and inter-PoP routes with primary interface E1A1 benefit from LFA link protection and node protection.

3.3.1.2. Per-Link LFA

We have a per-prefix LFA for A1 and hence we have a per-link LFA for link E1A1. All important intra-PoP and inter-PoP routes with primary interface E1A1 benefit from LFA per-link protection and de facto node protection.

3.3.2. A1E1 failure

3.3.2.1. Per-Prefix LFA

A1 has one single primary route via A1E1: the route to E1.

A1's LFA for route E1 is the path via A2 because eq1 == d < a + d. Node protection is not applicable.

Conclusion: all important routes with primary interface A1E1 benefit from LFA link protection. Node protection is not applicable.

3.3.2.2. Per-Link LFA

All important routes with primary interface A1E1 benefit from LFA link protection. De facto node protection is not applicable.

3.3.3. A1C1 failure

3.3.3.1. Per-Prefix LFA

Four destinations are impacted when A1C1 fails: C1, A3, E3, and P.

A1's LFA to C1 is via A2 because eq1 == u + c < a + u. Node protection property is not applicable for traffic to C1 when C1 fails.

A1's LFA to A3 is via A2 because eq1 == u + c + d < a + u + d. It is de facto node protecting as a < u + c + d (as we assumed a < u + d). Indeed A2 forwards traffic destined to A3 to C2, and C2 has a node protecting LFA for A3, for the failure of C2C1, being A4, as a < u + c + d. Hence the cascading application of LFAs by A1 and C2 during the failure of C1 provides de facto node protection.

A1's LFA to E3 is via A2 because eq1 == u + d + d < a + u + d + d. It is node protecting because eq2 == u + d + d < u + c + d + d.

A1's primary route to P is via C1 (even if e=0, u+x < u + c + x). The LFA is via A2 because eq1 == [u + c + x < a + u + x]. This LFA is node protecting (from the viewpoint of A1 computing eq2) if eq2 == u + x + e < u + c + x hence if e < c.

Conclusion: all important intra-PoP routes with primary interface A1C1 benefit from LFA link protection and node protection. Note that A3 benefits from a de facto node protection. All important inter-PoP routes with primary interface A1C1 benefit from LFA link protection. They also benefit from node protection if e < c.

3.3.3.2. Per-Link LFA

All important intra-PoP routes with primary interface A1C1 benefit from LFA link protection and de facto node protection. All important inter-PoP routes with primary interface A1C1 benefit from LFA link protection. They also benefit from de facto node protection if e < c.

3.3.4. C1A1 failure

3.3.4.1. Per-Prefix LFA

Three destinations are impacted by C1A1 link failure: A1, E1 and E2. E2's analysis is the same as E1 and hence is omitted.

C1's has no LFA for A1. Indeed, all its neighbors (C2 and A3) have a shortest path to A1 via C1. This is due to the assumption (c < a).

C1's LFA for E1 is via C2 because eq1 == d + d < c + d + d. It provides node protection because eq2 == d + d < d + a + d.

Conclusion: all important intra-PoP routes with primary interface A1C1 except A1 benefit from LFA link protection and node protection.

3.3.4.2. Per-Link LFA

C1 does not have a per-prefix LFA for destination A1 and hence there is no per-link LFA for the link C1A1.

3.3.4.3. Assumptions on the values of c and a

The commonly agreed design rule (c < a) is especially beneficial for a deployment using per-link LFA: it provides a per-link LFA for the most important direction (A1C1). Indeed, there are many more

destinations reachable over A1C1 than over C1A1. As the IGP convergence duration is proportional to the number of routes to update, there is a better benefit in leveraging LFA FRR for the link A1C1 than the link C1A1.

Note as well that the consequence of this assumption is much more important for per-link LFA than for per-prefix LFA.

For per-prefix LFA, in case of link C1A1 failure, we do have a per-prefix LFA for E1, E2 and any node subtended below A1 and A2. Typically most of the traffic traversing the link C1A1 is directed to these E nodes and hence the lack of per-prefix LFA for the destination A1 might be insignificant. This is a good example of the coverage benefit of per-prefix LFA over per-link LFA.

In the remainder of this section we analyze the consequence of not having c < a.

It definitely has a negative impact upon per-link LFA.

With c >= a, C1A1 has a per-link LFA while A1C1 has no per-link LFA. The number of destinations impacted by A1C1 failure is much larger than the direction C1A1 and hence the protection is provided for the wrong direction.

For per-prefix LFA, the availability of an LFA depends on the topology and needs to be assessed individually for each per-prefix. Some backbone topologies will lead to very good protection coverage, some others might provide very poor coverage.

More specifically, the coverage upon A1C1 failure of a remote destination P depends on whether e < a. In such case, A2 is a de-facto node-protecting per-prefix LFA for P.

Such a study likely requires a planning tool as each remote destination P would have a different e value (exception: all the edge devices of other aggregation pairs within the same region as for these e=0 by definition, e.g. E3).

Finally note that c = a is the worst choice as in this case C1 has no per-prefix LFA for A1 (and vice versa) and hence there is no per-link LFA for C1A1 and A1C1.

3.3.5. Conclusion

All important intra-PoP routes benefit from LFA link and node protection with one exception: C1 has no per-prefix LFA to A1.

All important inter-PoP routes benefit from LFA link protection. They benefit from node protection if e < c.

Per-link LFA provides the same protection coverage as per-prefix LFA with two exceptions. First, C1A1 has no per-link LFA at all. Second, when per-prefix LFA provides node protection (eq2 is satisfied), per-link LFA provides effective de facto node protection.

3.3.6. A square might become a full-mesh

If the vertical links of the square are made of parallel links (at L3 or at L2), then one should consider splitting these "vertical links" into "vertical and crossed links". The topology becomes "full-mesh". One should also ensure that the two resulting set of links (vertical and crossed) do not share any SRLG.

A typical reason preventing this is that the A1C1 bandwidth may be within a building while the A1C2 is between buildings. Hence while from a router port viewpoint the operation is cost-neutral, it is not from a cost of bandwidth viewpoint.

3.3.7. A full-mesh might be more economical than a square

In a full-mesh, the vertical and cross-links play the dominant role as they support most of the primary and backup paths. The capacity of the horizontal links can be dimensioned on the basis of traffic destined to a single C or a single A and a single E node.

3.4. Extended U

For the Extended U topology, we define the following terminology:

C1L1: the node "C1" as seen in topology L1.

C1L2: the node "C1" as seen in topology L2.

C1LO: the loopback of C1. This loopback is in L2.

C2LO: the loopback of C2. This loopback is in L2.

Let us also remind that C1 and C2 are L1L2 routers and that their loopbacks are in L2 only.

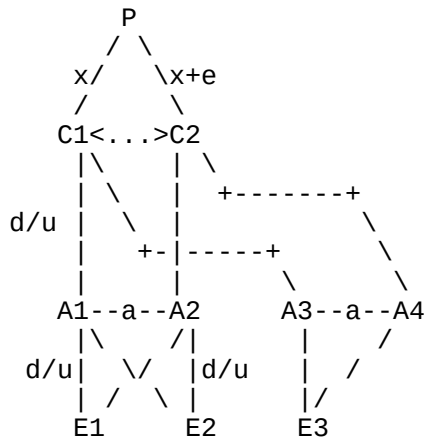


Figure 5: Extended U

There is no L1 link between C1 and C2. There might be an L2 link between C1 and C2. This is not relevant as this is not seen from the viewpoint of the L1 topology which is the focus of our analysis.

It is guaranteed that there is a path from C1LO to C2LO within the L2 topology (except if the L2 topology partitions which is very unlikely and hence not analyzed here). We call "c" its path cost. Once again, we assume that c < a.

We exploit this property to create a tunnel T between C1LO and C2LO. Once again, as the source and destination addresses are the loopbacks of C1 and C2 and these loopbacks are in L2 only, it is guaranteed that the tunnel does not transit via the L1 domain.

IS-IS does not run over the tunnel and hence the tunnel is not used for any primary paths within the L1 or L2 topology.

Within Level1, we configure C1 (C2) with a Level1 LFA extended neighbor "C2 via tunnel T" ("C1 via tunnel T").

A router supporting such extension learns that it has one additional potential neighbor in topology Level1 when checking for LFA’s.

The L1 topology learns about C1LO as an L2=>L1 route with Down bit set propagated by C1L1 and C2L1. The metric advertised by C2L1 is bigger than the metric advertised by C1L1 by "c".

The L1 topology learns about P as an L2=>L1 routes with Down bit set propagated by C1L1 and C2L1. The metric advertised by C2L1 is bigger than the metric advertised by C1L1 by "e". This implies that e <= c.

3.4.1. E1A1 failure

3.4.1.1. Per-Prefix LFA

Five destinations are impacted by E1A1 link failure: A1, C1LO, E2, E3 and P.

The LFA for A1 is via A2 because eq1 == a < d + u. Node protection for traffic to A1 upon A1 node failure is not applicable.

The LFA for E2 is via A2 because eq1 == d < d + u + d. Node protection is guaranteed because eq2 == d < a + d.

The LFA for E3 is via A2 because eq1 == u + d + d < d + u + d + d. Node protection is guaranteed because eq2 == u + d + d < a + u + d + d.

The LFA for C1LO is via A2 because eq1 == u + c < d + u + u. Node protection is guaranteed because eq2 == u + c < a + u.

If e=0: E1’s primary route to P is via ECMP(E1A1, E1A2). The LFA for the first ECMP path (via A1) is the second ECMP path (via A2). Node protection is possible because eq2 == u + x < a + u + x.

If e<>0: E1’s primary route to P is via E1A1. Its LFA is via A2 because eq1 == a + c + x < d + u + u + x. Node protection is guaranteed because eq2 == u + x + e < a + u + x <=> e < a. This is true because e <= c and c < a.

Conclusion: same as the square topology.

3.4.1.2. Per-Link LFA

Same as the square topology.

3.4.2. A1E1 failure

3.4.2.1. Per-Prefix LFA

Same as the square topology.

3.4.2.2. Per-Link LFA

Same as the square topology.

3.4.3. A1C1 failure

3.4.3.1. Per-Prefix LFA

Three destinations are impacted when A1C1 fails: C1, E3 and P.

A1's LFA to C1LO is via A2 because eq1 == u + c < a + u. Node protection property is not applicable for traffic to C1 when C1 fails.

A1's LFA to E3 is via A2 because eq1 == u + d + d < d + u + u + d + d. Node protection is guaranteed because eq2 == u + d + d < a + u + d + d.

A1's primary route to P is via C1 (even if e=0, u + x < a + u + x). The LFA is via A2 because eq1 == u + x + e < a + u + x <=> e < a (which is true see above). Node protection is guaranteed because eq2 == u + x + e < a + u + x.

Conclusion: same as the square topology

3.4.3.2. Per-Link LFA

Same as the square topology.

3.4.4. C1A1 failure

3.4.4.1. Per-Prefix LFA

Three destinations are impacted by C1A1 link failure: A1, E1 and E2. E2's analysis is the same as E1 and hence is omitted.

C1L1 has an LFA for A1 via the extended neighbor C2L1 reachable via tunnel T. Indeed, eq1 is true: d + a < d + a + u + d. From the viewpoint of C1L1, C2L1's path to C1L1 is C2L1-A2-A1-C1L1. Remember the tunnel is not seen by IS-IS for computing primary paths! Node protection is not applicable for traffic to A1 when A1 fails.

C1L1's LFA for E1 is via extended neighbor C2L1 (over tunnel T) because eq1 == d + d < d + a + u + d + d. Node protection is guaranteed because eq2 == d + d < d + a + d.

3.4.4.2. Per-Link LFA

C1 has a per-prefix LFA for destination A1 and hence there is a per-link LFA for the link C1A1. Node resistance is applicable for traffic to E1 (and E2).

3.4.5. Conclusion

The extended U topology is as good as the square topology.

It does not require any cross links between the A and C nodes within an aggregation region. It does not need an L1 link between the C routers in an access region. Note that a link between the C routers might exist in the L2 topology.

3.5. Dual-plane Core and its impact on the Access LFA analysis

A Dual-plane core is defined as follows
- Each access region k is connected to the core by two C routers (C(1,k) and C(2,k)).
- C(1,k) is part of Plane1 of the dual-plane core.
- C(2,k) is part of Plane2 of the dual-plane core.
- C(1,k) has a link to C(2, l) iff k = l
- {C(1,k) has a link to C(1, l)} iff {C(2,k) has a link to C(2, l)}

In a dual-plane core design, e = 0 and hence the LFA node-protection coverage is improved in all the analyzed topologies.

3.6. Two-tiered IGP metric allocation

A Two-tiered IGP metric allocation scheme is defined as follows
- all the link metrics used in the L2 domain are part of range R1
- all the link metrics used in an L1 domain are part of range R2
- range R1 << range R2 such that the difference e = C2P - C1P is smaller than any link metric within an access region.

Assuming such an IGP metric allocation, the following properties are guaranteed : c < a, e < c, and e < a.

3.7. uLoop analysis

In this section, we analyze a case where the routing transition following the failure of a link may have some uLoop potential for one destination. Then we show that all the other cases do not have uLoop potential.

In the square design, upon the failure of link C1A1, traffic addressed to A1 can undergo a transient forwarding loop as C1 reroutes traffic to C2, which initially reaches A1 through C1, as c < a. This loop will actually occur when C1 updates its FIB for destination A1 before C2.

It can be shown that all the other routing transitions following a link failure in the analyzed topologies do not have uLoop potential.

Indeed, in each case, for all destinations affected by the failure, the rerouting nodes deviate their traffic directly to adjacent nodes whose paths towards these destinations do not change. As a consequence, all these routing transitions cannot undergo transient forwarding loops.

For example, in the square topology, the failure of directed link A1C1 does not lead to any uLoop. The destinations reached over that directed link are C1 and P. A1 and E1's shortest paths to these destinations after the convergence go via A2. A2's path to C1 and P is not using A1C1 before the failure, hence no uLoop may occur.

3.8. Summary

In this section, we summarize the applicability of LFAs detailed in the previous sections. For link protection, we use "Full" to refer to the applicability of LFAs for each destination, reached via any link of the topology. For node protection, we use "yes" to refer to the fact that node protection is achieved for a given node.

1. Intra Area Destinations
   - Link Protection
     - Triangle: Full
     - Full-Mesh: Full
     - Square: Full, except C1 has no LFA for dest A1
     - Extended U: Full
   - Node Protection
     - Triangle: yes.
     - Full-Mesh: yes.
     - Square: yes.
     - Extended U: yes.
2. Inter Area Destinations
   - Link Protection
     - Triangle: Full
     - Full-Mesh: Full
     - Square: Full
     - Extended U: Full
   - Node Protection
     - Triangle: yes if e<c
     - Full-Mesh: yes for A failure, if e<c for C failure
     - Square: yes for A failure, if e<c for C failure
     - Extended U : yes if e<= c and c < a
3. uLoops
   - Triangle: None
   - Full-Mesh: None
   - Square: None, except traffic to A1 when C1A1 fails
   - Extended U : None, if a > e

4. Per-Link LFA vs Per-Prefix LFA
   * Triangle: Same
   * Full-Mesh: Same
   * Square: Same except C1A1 has no per-Link LFA. In practice, this means that per-prefix LFAs will be used (hence C1 has no LFA for dest=E1 and dest=A1)
   * Extended U : Same

4. Core Network

In the backbone, the optimization of the network design to achieve the maximum LFA protection is less straightforward than in the case of the access/aggregation network.

The main optimization objectives for backbone topology design are cost, latency, and bandwidth, constrained by the availability of fiber. Optimizing the design for Local IP restoration is more likely to be considered as a non-primary objective. For example, the way the fiber is laid out and the resulting cost to change it leads to ring topologies in some backbone networks.

Also, the capacity planning process is already complex in the backbone. It needs to make sure that the traffic matrix (demand) is supported by the underlying network (capacity) under all possible variation of the underlying network (what-if scenario related to one-srlg failure). Classically, "supported" means that no congestion be experienced and that the demands be routed along the appropriate latency paths. Selecting LFA as a deterministic FRR solution for the backbone would require to enhance the capacity planning process to add a third constraint: each variation of the underlying network should lead to a sufficient LFA coverage (we detail this aspect in a following section).

To the contrary, the access network is based on many replications of a small number of well-known (well-engineered) topologies. The LFA coverage is deterministic and is independent of additions/insertions of a new edge device, a new aggregation sub-region or a new access region.

In practice, we believe that there are three profiles for the backbone applicability of LFA.

In the first profile, the designer plans all the network resilience on IGP convergence. In such case, LFA is a free bonus. If an LFA is available, then the loss of connectivity is likely reduced by a factor 10 (50msec vs 500msec), else the loss of connectivity depends on IGP convergence which is anyway the initial target. LFA should be

very successful here as it provides a significant improvement without any additional cost.

In the second profile, the designer seeks a very high and deterministic FRR coverage and he either does not want or cannot engineer the topology. LFA should not be considered in this case. MPLS TE FRR would perform much better in this environment. Explicit routing ensures that a backup path exists what-ever the underlying topology.

In the third profile, the designer seeks a very high and deterministic FRR coverage and he does engineer the topology. LFA is appealing in this scenario as it can provide a very simple way to obtain protection. Furthermore, in practice, the requirement for FRR coverage might be limited to a certain part of the network, given by a sub-topology and/or is likely limited to a subset of the demands within the traffic matrix. In such case, if the relevant part of the network natively provides a high degree of LFA protection for the demands of interest, it might actually be straightforward to improve the topology and achieve the level of protection required for the sub-topology and demands which matter. Once again, the practical problem needs to be considered (which sub-topology, which real demands need 50msec) as it is often simpler than the theoretical generic one.

For the reasons explained previously, the backbone applicability should be analyzed on a case by case basis and it is difficult to derive generic rules.

In order to help the reader to assess the LFA applicability in its own case, we provide in the next section some simulation results based on 11 real backbone topologies.

4.1. Simulation Framework

In order to perform an analysis of LFA applicability in the core, we usually receive the complete IS-IS/OSPF linkstate database taken on a core router. We parse it to obtain the topology. During this process, we eliminate all nodes connected to the topology with a single link and all prefixes except a single "node address" per router. We compute the availability of per-prefix LFA's to all these node addresses which we call "destinations" hereafter. We treat each link in each direction.

For each (directed) link, we compute whether we have a per-prefix LFA to the next-hop. If so, we have a per-link LFA for the link.

The Per-link-LFA coverage for a topology T is the fraction of the

number of links with a per-link LFA divided by the total number of links.

For each link, we compute the number of destinations whose primary path involves the analyzed link. For each such destination, we compute whether a per-prefix LFA exists.

The Per-Prefix-LFA coverage for a topology T is the fraction:

(the sum across all links of the number of destinations with a primary path over the link and a per-prefix LFA)

divided by

(the sum across all links of the number of destinations with a primary path over the link)

4.2. Data Set

Our data set is based on 11 SP core topologies with different geographical scopes: worldwide, national and regional. The number of nodes range from 600 to 16. The average link-to-node ratio is 2.3 with a minimum of 1.2 and maximum of 6.

4.3. Simulation results

Topology	Per-link LFA	Per-prefix LFA
T1	45%	76%
T2	49%	98%
T3	88%	99%
T4	68%	84%
T5	75%	94%
T6	87%	98%
T7	16%	67%
T8	87%	99%
T9	67%	79%
T10	98%	99%
T11	59%	77%
Average	67%	89%
Median	68%	94%

Table 1: Core LFA Coverages

In Table 1, we observe a wide variation in terms of LFA coverage across topologies; From 67% to 100% for the per-prefix LFA coverage,

and from 16% to 98% for the per-link LFA coverage. Several topologies have been optimized for LFAs (T3, 6, 8 and 10). This illustrates the need for case by case analysis when considering LFA for core networks.

It should be noted that, to the contrary of the access/aggregation topologies, per-prefix LFA outperforms per-link LFA in the backbone.

5. Core and Access protection schemes are independent

Specifically, a design might use LFA FRR in the access and MPLS TE FRR in the core.

LFA provides great benefits for the access network due to its excellent access coverage and its simplicity.

MPLS TE FRR's topology independence might prove beneficial in the core when either the LFA FRR coverage is judged too small and/or the designer feels unable to optimize the topology to improve the LFA coverage.

6. Simplicity and other LFA benefits

The LFA solution provides significant benefits which mainly stem from its simplicity.

The LFA behavior is an automated process that makes fast restoration an intrinsic part of the IGP, with no additional configuration burden in the IGP or any other protocol.

Thanks to this integration, the use of multiple areas in the IGP does not make Fast Restoration more complex to achieve than in a single area IGP design.

There is no requirement for network-wide upgrade as LFAs do not require any protocol change and hence can be deployed router by router.

With LFAs, the backup paths are pre-computed and installed in the dataplane in advance of the failure. Assuming a fast enough FIB update time compared to the total number of (important) destinations, a "<50msec repair" requirement becomes achievable. With a prefix-independent implementation, LFAs have a fixed repair time, as it only depends on the failure detection time and the time to activate the LFA behavior, which does not scale with the number of destinations to be fast rerouted.

Link and node protection are provided together and without operational difference (as a comparison, MPLS TE FRR link and node protections require different types of backup tunnels and different grades of operational complexity).

Also, compared to MPLS TE FRR, an important simplicity aspect of LFA is that is does not require the introduction of yet another virtual layer of topology. Maintaining a virtual topology of explicit MPLS TE tunnels clearly increases the complexity of the network. MPLS TE tunnels would have to be represented in a network management system in order to be monitored and managed. In large networks this may significantly contribute to the number of network entities polled by the network management system and monitored by operational staff. LFA on the other hand only has to be monitored for its operational status once per router and it needs to be considered in the network planning process. If the latter is done based on offline simulations for failure cases anyways, the incremental cost of supporting LFA for a defined set of demands may be relatively low.

The per-prefix mode of LFAs allows for a simpler and more efficient capacity planning. As the backup path of each destination is optimized individually, the load to be fast rerouted can be spread on a set of shortest-repair-paths (as opposed to one single backup tunnel). This leads for a simpler and more efficient capacity planning process that takes congestion during protection into account.

7. Capacity Planning with LFA in mind

We briefly describe the functionality a designer should expect from a capacity planning tool supporting LFA and the related capacity planning process.

7.1. Coverage Estimation - Default Topology

Per-Link LFA Coverage Estimation: the tool would color each unidirectional link in depending on whether per-link LFA is available or not. Per-Prefix LFA Coverage Estimation: the tool would color each unidirectional link with a colored gradient based on the % of destinations which have a per-prefix LFA.

On top of the visual GUI reporting, the tool should provide detailed tables listing, on a per interface basis: percentage of LFA, number of prefixes with LFA, number without LFA, list of prefixes without LFA.

Furthermore, the tool should provide the percentage and list the

traffic matrix demands with less than 100% source-to-destination LFA coverage, and, average coverage (#links this demand has an LFA on/# links this demands traverses) for every demands (using a threshold).

The user should be able to alter the color scheme to show whether these LFAs are guaranteed-node-protecting or de-facto node protecting or only link protecting.

This functionality provides the same level of information as we described in sections 4.1 to 4.3.

7.2. Coverage estimation in relation to traffic

Instead of reporting the coverage as a ratio of the number of destinations with a backup, one might prefer a ratio of the amount of traffic on a link that benefits from protection.

This is likely much more relevant as not all destinations are equal and it is much more important to have an LFA for a destination attracting lots of traffic rather than an unpopular destination.

7.3. Coverage verification for a given set of demands

Depending on the requirements on the network it might be more relevant to verify the complete LFA coverage of a given sub-topology, or a given set of demands, rather than calculating the relative coverage of the overall traffic. This is most likely true for the third engineering profile described in Section 4.

In that case, the tool should be able to separately report the LFA coverage on a given set of demands and highlight each part of the network that does not support 100% coverage for any of those demands.

7.4. Modeling - What-if Scenarios - Coverage impact

The tool should be able to compute the coverage for all the possible topologies that result from a set of expected failures (ie. one-srlg failure).

Filtering the key information from the huge amount of generated data should be a key property of the tool.

For example, the user could set a threshold (at least 80% per-prefix LFA coverage in all one-srlg what-if scenarios) and the tool would report only the cases where this condition is not met, hopefully with some assistance on how to remedy the problem (IGP metric optimization).

As an application example, a designer who is not able to ensure c < a could leverage such a tool to assess the per-prefix LFA coverage for square aggregation topologies grafted to its core backbone topology. The tool would analyze the per-prefix LFA availability for each remote destination and would help optimize the backbone topology to increase the LFA protection coverage for failures within the square aggregation topologies.

7.5. Modeling - What-if Scenarios - Load impact

The tool should be able to compute the link load for all routing states that result from a set of expected failures (i.e. one-srlg failure).

The routing states that should be supported are: 1/ network-wide converged state before the failure, 2/ all the LFA's protecting the failure are active and 3/ network-wide converged state after the failure.

Filtering the key information from the huge amount of generated data should be a key property of the tool.

For example, the user could set a threshold (at most 100% link load in all one-srlg what-if scenarios) and the tool would report only the cases where this condition is violated, hopefully with some assistance on how to remedy the problem (IGP metric optimization).

The tool should be able to do this for the aggregate load and as well on a per class of service basis.

Note: in case the traffic matrix is unknown, an intermediate solution consists in identifying the destinations that would attract traffic (i.e. PE routers), and those that would not (i.e. P routers). You could achieve this by creating a traffic matrix with equal demands between the sources/destinations that would attract traffic (Pe to PE). This will be more relevant than considering all demands between all prefixes (e.g. when there is no customer traffic from P to P).

7.6. Discussion on metric recommendations

While LFA FRR has many benefits (section 6), LFA FRR's applicability depends on topology.

The purpose of this document is to show how to introduce a level of control on this topology parameter.

On the one hand, we wanted to show that by adopting a small set of igp metric constraints and a repetition of well-behaved patterns, the

designer could deterministically guarantee maximum link and node protection for the vast majority of the network (the access/ aggregation). Doing so, he would obtain an extremely simple resiliency solution.

One another side, we also wanted to show that it might not be so bad to not apply (all) these constraints.

Indeed, we showed in section 3.3.4.3 that the per-prefix LFA coverage in a square where c > a might still be very good.

We showed in section 4.3 that the median per-prefix LFA coverage for 11 SP backbone topologies still provides for 94% coverage (most of these topologies were built without any idea of LFA)!

Furthermore, we showed that any topology may be analyzed with an LFA-aware capacity planning tool. This would readily assess the coverage of per-prefix LFA and would assist the designer in fine-tuning it to obtain the level of protection he seeks.

While this document highlighted LFA applicability and benefits for SP network, it also noted that LFA is not meant to replace MPLS TE FRR.

With a very-LFA-unfriendly topology, a designer seeking a guaranteed < 50msec protection might be better off leveraging the explicit-routed backup capability of MPLS TE FRR to provide 100% protection while ensuring no congestion along the backup paths during protection.

But when LFA provides 100% link and node protection without any uLoop, then clearly LFA seems a technology to consider to drastically simplify the operation of a large-scale network.

8. Security Considerations

The security considerations applicable to LFAs are described in [RFC5286]. This document does not introduce any new security considerations.

9. IANA considerations

This draft does not require any IANA considerations.

10. Conclusions

LFA is an important protection alternative for IP/MPLS networks.

Its simplicity benefit is significant, in terms of automation and integration with the default IGP behavior and the absence of any requirement for network-wide upgrade. The technology does not require any protocol change and hence can be deployed router by router.

At first sight, these significant simplicity benefits are negated by the topological dependency of its applicability.

The purpose of this document was to highlight that very frequent access and aggregation topologies benefit from excellent link and node LFA coverage.

A second objective consisted in describing the three different profiles of LFA applicability for the IP/MPLS core networks and illustrating them with simulation results based on real SP core topologies.

11. Contributors

This work has been realized in tight collaboration with the following people.

Mike Shand
imc.shand@googlemail.com

Bruno Decraene
France Telecom
38-40 rue du General Leclerc
92794 Issy Moulineaux cedex 9
FR
bruno.decraene@orange.com

James Uttaro
ATT
200 S. Laurel Avenue
07748, Middletown, NJ
US
uttaro@att.com

Nicolai Leymann

Deutsche Telekom
Winterfeldtstrasse 21
10781, Berlin
DE
N.Leymann@telekom.de

Martin Horneffer
Deutsche Telekom
Hammer Str. 216-226
48153, Muenster
DE
Martin.Horneffer@telekom.de

12. Acknowledgments

We would like to thank Alvaro Retana and Stewart Bryant (in bold) for their precious comments on this work.

13. References

13.1. Normative References

[RFC5286] Atlas, A. and A. Zinin, "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, September 2008.

13.2. Informative References

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, January 2010.

[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", RFC 2328, April 1998.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.

Authors' Addresses

Clarence Filsfils
Cisco Systems
Brussels 1000
BE

Email: cf@cisco.com

Pierre Francois
Institute IMDEA Networks
Avda. del Mar Mediterraneo, 22
Leganese 28918
ES

Email: pierre.francois@imdea.org

This Internet-Draft, draft-karan-mofrr-01.txt, has expired, and has been deleted from the Internet-Drafts directory. An Internet-Draft expires 185 days from the date that it is posted unless it is replaced by an updated version, or the Secretariat has been notified that the document is under official review by the IESG or has been passed to the RFC Editor for review and/or publication as an RFC. This Internet-Draft was not published as an RFC.

Internet-Drafts are not archival documents, and copies of Internet-Drafts that have been deleted from the directory are not available. The Secretariat does not have any information regarding the future plans of the authors or working group, if applicable, with respect to this deleted Internet-Draft. For more information, or to request a copy of the document, please contact the authors directly.

Draft Authors:
Apoorva Karan<apoorva@cisco.com>
Clarence Filsfils<cfilsfil@cisco.com>
Dino Farinacci<dino@cisco.com>
Bruno Decraene<bruno.decraene@orange-ftgroup.com>
Nicolai Leymann<n.leymann@telekom.de>
Thomas Telkamp<telkamp@cariden.com>

RTGWG
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2011

W. Lu
A. Tian
S. Kini
Ericsson

October 18, 2010

Fast Notification Framework
draft-lu-fast-notification-framework-00

Abstract

This document describes an architectural work that competes with the IP Fast Re-Route (IPFRR) solution which aims to minimize the network down time in the event of equipments failure. The work provides a layered framework based upon which applications such as the domain-wide fast convergence may be achieved through the transport layer fast delivery of failure notifications.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 21, 2011.

Table of Contents

1. Introduction

The ability to recover rapidly from network failures is one of the most sought network characteristics. Few solutions address this issue to the satisfactory.

IPFRR [RFC5714] is one such solution. It mimics MPLS-FRR [RFC4090] solution. The difference is that the MPLS-FRR is path based, or source routing based in other words. This implies that the re-route decision can be carried out by the PLR (point-of-local-repair) router alone, with no need of cooperation of other LSRs in the network.

Unfortunately, IP based FRR is by nature not source routing based. Its re-route decision may not be honored by other routers in the network. The consequence can be very severe, either traffic outage or even routing loops.

Many methods were proposed around IPFRR concept but none is close to be satisfactory. Some methods such as LFA described in [RFC5286] require lot of computation and have coverage issue. Some others such as Not-Via [I-D.ietf-rtgwg-ipfrr-notvia-addresses] are extremely complicated and are prohibitive to be useful.

The primary reason for such difficulties can be understood from the following passage which is quoted from [RFC5714] first paragraph of section 1:

However, there is an alternative approach, which is to compute backup routes that allow the failure to be repaired locally by the router(s) detecting the failure without the immediate need to inform other routers of the failure.

The phrase "without the immediate need to inform other routers of the failure" is against the very nature of the IP network in which the domain-wide synchronization is the key.

In this document we propose a method which directly addresses the rapid network synchronization needs. It is not IPFRR based. However it can achieve the same or better result without much complexity and compromise.

The method lays out a framework which decouples the improvement in the forwarding plane from the control plane. The design also allows and promotes future innovations based upon the framework.

1.1. Requirements Language

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

1.2. Acronyms

FRR - Fast Re-Route

IPFRR - IP Fast Re-Route

MPLS - Multi-Protocol Label Switch

LFA - Loop Free Alternative

TLV - Type Length Value tuple

IGP - Interior Gateway protocol

OSPF - Open Shortest Path First

IS-IS - Intermediate System to Intermediate System

PDU - Protocol Data Unit

DoS - Denial of Service

FNF - Fast Notification Framework

2. Event Framework

An event framework is introduced for the purpose of rapid disseminating of events to all interested receivers in a network.

The framework is application independent. Many applications can generate the events and/or register to receive the events. A TLV based framework is proposed to ensure separation between application and the delivery framework.

The event framework is also independent of the underlying delivery mechanisms. Different delivery mechanisms may be introduced, each with different properties suitable for different requirements. For example, some delivery mechanism is solely optimized for simplicity; while other may improve on reliability.

One of the use cases of this event framework is Fast Failure

Notification, which can be used to improve network convergence time. When a failure occurs in a network, routers adjacent to the failure can detect it and quickly disseminate the failure notifications to other routers throughout the area. Routing protocols on different routers can register and receive such failure notifications, then quickly react to the failure to achieve fast convergence.

The routing protocols discussed in this work are Interior Gateway Protocols (IGP) with the focus on the Link State Routing Protocols such as Open Shortest Path First [RFC2328] and Intermediate System to Intermediate System [RFC1195] [ISO.10589.1992].

The event in the scope of this architecture is specifically the link-down event or node-down event. The up events are not fast flooded for the sake of network stability.

3. Layered Structure

The framework can be viewed as a layered structure in which various routing functions can be rearranged. This arrangement is based on the principle of separation of functions. It will facilitate the innovation in various component building blocks and in the mean while allow them to integrate in a systematic manner.

There are two layers that make the framework. One is for routing protocol specific functionality. The other is the data transport layer. Figure 1 depicts this concept.

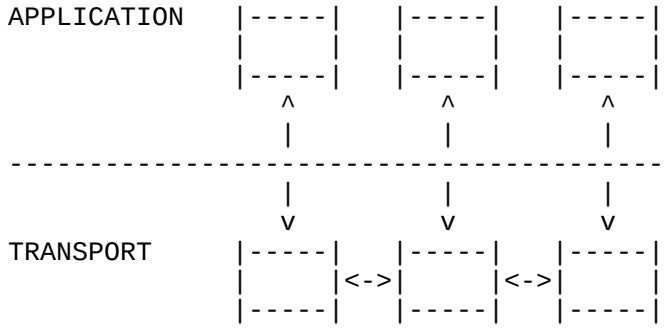


Figure 1: Fast Notification Architecture

Regular routing protocol performs the flooding in store-and-forward manner. While this is reliable (retransmission) and secure (adjacency check), it involves control plane operation and the control plane to data plane communication. It inevitably drags the

network-wide convergence.

With the fast notification architecture, the delivery function is detached from the application layer and moved onto the transport layer. More precisely, the transport layer provides domain-wide fast delivery platform. The normal flooding function is still kept in the application layer to ensure ultimate synchronization in case the fast flooding does not reach some intended routers for whatever reasons.

The speed of the fast flooding needs not to be faster than the data traffic. As long as the messenger travels at the same speed of the data traffic, it always gives the next-hop router the same amount of time for processing as it gives the previous router.

4. Operation

Fast failure notification operates on following steps:

1. Failure detection;

2. Notification composing and dispatching;

3. Notification flooding;

4. Notification receiving;

5. Routing/forwarding table update.

4.1. Failure detection

This can be made in many ways. But it has to be fast and light-weight. Layer-2 link-event monitoring and signaling is obvious an option. Bidirectional Forwarding detection (BFD) is also a good candidate. There may be more, or combinations of them.

The fast notification architecture encourages the innovation in this area which can be pursued freely and independently.

4.2. Notification Origination

This part involves the message format. This document does not specify or endorse a particular format. It is open to any format as long as it fulfills the fast flooding purpose. The detecting router is responsible for the initiation of the fast notification process. Its action is the starting point of the fast flooding.

There are two packet formats worth of mentioning.

4.2.1. IGP PDU

The simplest approach is to use the IGP packet format directly. For example, the OSPF Router-LSA packet which reflects a broken adjacency (one fewer router link) can be fast-flooded to all routers without special modification.

The benefit is that the receivers can process the packet as usual. Moreover since the packet is no different than the one in normal flooding, it guarantees the seamless transition when the "slow" flooding catches up. Plus, there will be no duplicate effort of fast and slow convergence. Flooding stops wherever a router is updated (already fast flooded).

The drawback is that the message cannot be made uniform for multiple protocols. Other protocol such as IS-IS will have to devise a different format. In addition, since IS-IS PDU is not IP based, it may require encapsulation in some cases.

Another drawback is that the normal IGP flooding uses adjacency check to prevent DoS attack or PDU replay from un-trusted parties. The check has to be bypassed for the fast-flooded packets to be accepted. This opens door to the DoS or some other attacks. Domain-wide authentication may be adopted for protection.

4.2.2. Uniform Message

This format must include essential and sufficient information about the broken link. The message will be treated on the receiver router as a local event. The uniformed messaging provides freedom for future expansion. The format thus is recommended TLV-based.

Cautions must be taken in case the message is mistakenly flooded due to bugs or some error conditions. Timeout machinery may be used to protect against such issues.

The detecting router is responsible for the initiation of the fast notification process. Its action is the starting point of the fast flooding.

4.3. Fast Flooding

The fast flooding does not specify the fast flooding mechanism. It is up to the routing society to figure out and single out good solutions. The requirement is that the flooding has to be

a. Reliable in that it reaches all participants even after failures occur;

b. Loop-free;

c. Simple;

d. Can be authenticated.

4.4. Notification Receiving and Handling

This involves upon the arrival of the notification message, how it is forwarded to the routing protocol for further processing. If the fast-flooding scheme uses specific IP destination addresses or MAC addresses, the receiving router has to recognize it.

When the message reaches the protocol process, it may have to relax its acceptance criteria.

If in the future, some algorithm is developed that the notification handling takes very few CPU cycles, this process may be performed in real-time. Therefore it is worthy of considering move the notification handling into the data plane. This will cut a large chunk of delay and may lead to hitless domain-wide convergence.

4.5. Routing/Forwarding Table Update

This should be the same as normal IGP decision process. It is also possible to pre-download the changes to the data plane if the complexity can be limited. This will improve the overall convergence time dramatically.

5. Convergence Analyses

5.1. Definition of Convergence Time

The convergence time is measured by dividing the number of lost packets with the traffic flow rate between any two routers in the domain. This SHOULD equal to the domain wide network convergence time if all individual routers have the same computing power and the same convergence time.

5.2. Domain Wide Convergence

Due to the propagation delay, all routers do not converge at the same time. The traffic loss, however, stops immediately after the first router repairs.

This is because the data traffic has to go through the same propagation delay, which exactly compensate the late starting of the convergence at remote routers.

Take a ring topology for example, as shown in Figure 2.

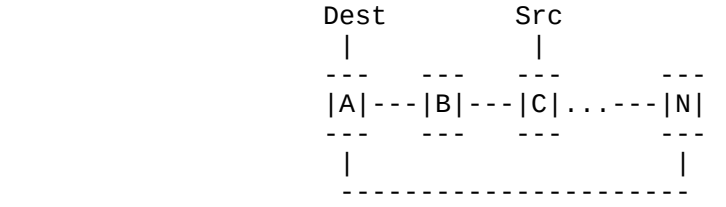


Figure 2: Ring Topology

Assume all routers have same convergence time 50 milliseconds. Assume the transmission delay over each hop is 20 milliseconds.

Upon link A-B failure, B floods its Link State Update to C. Table 1 shows the convergence timeline.

Node	Converge Starts	converge Completes
B	0	50ms
C	20ms	70ms

During the first 50 milliseconds, packets from B to A are dropped. Right after 50th milliseconds, B re-routes packets toward C. Those packets, after traveling 20 milliseconds, arrive C at 70th milliseconds when C is just repaired. Since C and all downstream routers will correct themselves one by one right before those packets arrive, they will arrive at the destination via the corrected path successfully. The overall convergence time is thus same as B's.

5.3. Micro-looping

If routers' convergence time is different, micro looping may form, although packets will still be delivered after several loops. Still use Figure 2 for example. Assume C needs 90 milliseconds to converge. When B re-routes packets back to C at 70th milliseconds, C has not finished its updating yet. It continues to use its old forwarding table and bounces packets back to B. B in turn re-route packets again to C. This time packets arrive at C at 110th milliseconds. C has done updating and will forward packets

correctly. The packets are looped once.

The micro-looping does not form easily with Fast Flooding method. The routers have to differ in computing speed and differ significantly.

5.4. Packet Reordering

Due to the different convergence timeline, packets may be temporarily forwarded in wrong direction before being placed on the right track. This will not cause packet loss, but will result in packet reordering.

Packet reordering affects TCP communication adversely in that new sequence numbered packets may arrive ahead of the older ones.

This problem is common in IPFRR solutions, and remains an open issue. Not-Via for example, may have packets reordered when it switches to use the final stable routes from the temporary LFAs. On the other hand, the connectionless network by nature never promises ordered packet delivery. This type of problem deserves a separate topic and is beyond the scope of this document.

6. Scalability Analyses

Fast Flooding scales with networks of any size and any topology. At least it scales no inferior to the normal IGP flooding.

7. Traffic Analyses

Traffics that did not route through the broken link are intact. Traffics that did will be successfully re-routed as soon as the affected router converges (as opposed to all routers converge).

Upon the convergence of the affected router, Fast Flooding guarantees correct routes for all affected traffics.

8. Acknowledgements

TBD

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

TBD

11. References

11.1. Normative References

[ISO.10589.1992]
International Organization for Standardization, "Intermediate system to intermediate system intra-domain-routing routine information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", ISO Standard 10589, 1992.

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

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

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

11.2. Informative References

[I-D.ietf-rtgwg-ipfrr-notvia-addresses]
Shand, M., Bryant, S., and S. Previdi, "IP Fast Reroute Using Not-via Addresses", draft-ietf-rtgwg-ipfrr-notvia-addresses-05 (work in progress), March 2010.

[RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.

[RFC5286] Atlas, A. and A. Zinin, "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, September 2008.

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, January 2010.

Authors' Addresses

Wenhu Lu
Ericsson
300 Holger Way
San Jose, California 95134
USA

Phone: 408 750-5436
Email: wenhu.lu@ericsson.com

Albert Tian
Ericsson
300 Holger Way
San Jose, California 95134
USA

Phone: 408 750-8739
Email: albert.tian@ericsson.com

Sriganesh Kini
Ericsson
300 Holger Way
San Jose, California 95134
USA

Phone: 408 750-5210
Email: sriganesh.kini@ericsson.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 21, 2014

W. Lu
S. Kini
A. Csaszar, Ed.
G. Enyedi
J. Tantsura
Ericsson

August 20, 2013

Transport of Fast Notification Messages
draft-lu-fn-transport-05

Abstract

This document specifies mechanisms for fast and light-weight dissemination of event notifications. The purpose is to enable dataplane dissemination of Fast Notifications (FNs). The draft discusses the design goals, the message container and options for delivering the notifications to all routers within a routing area.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 21, 2014.

Copyright Notice

Table of Contents

1. Introduction

Enabling fast dissemination of a network event to routers in a limited area could benefit multiple applications. Existing use cases

are centered around new approaches for IP Fast ReRoute such as [I-D.csaszar-ipfrr-fn]. In the future, however, multiple innovative applications may take advantage of a Fast Notification service.

A hop by hop control plane based flooding mechanism is used widely today in link state routing protocols such as OSPF and ISIS to propagate routing information throughout an area. In this mechanism, the information is processed in the control plane at each hop before being forwarded to the next. The extra processing, scheduling, and communications overhead causes unnecessary delays in the dissemination of the information.

This draft proposes a generic fast notification (FN) protocol as a separate transport layer, which focuses on delivering notifications quickly in a secure manner. It can be used by many existing applications to enhance the performance of those applications, as well as to enable new services in the network. This draft does not specify the payload of the notification. Each application is required to create an own spec and define its payload as well as the preferred transport options separately.

1.1. Requirements Language

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

1.2. Acronyms

FN - Fast Notification

IGP - Interior Gateway Protocol

IS-IS - Intermediate System to Intermediate System

MD5 - Message Digest 5

OSPF - Open Shortest Path First

RPF - Reverse Path Forwarding

SHA - Secure Hash

SPT - Shortest Path Tree

STP - Spanning Tree Protocol

2. Design Goals

A light-weight event notification mechanism that could be used to facilitate quick dissemination of information in a limited area should have the following properties.

1. The mechanism should be fast. It should provide low end to end propagation delay for the notifications.

2. The signaling mechanism should offer a high degree of reliability under network failure conditions.

3. The mechanism should be secure; that is, it should provide means to verify the authenticity of the notifications.

4. The new protocol should not be dependent upon routing protocol flooding procedures.

5. The mechanism should have low processing overhead.

These design goals present a trade-off. Proper balance needs to be found that offers good authentication and reliability while keeping processing complexity sufficiently low to enable implementation in dataplane. This draft proposes solutions that take the above goals and trade-offs into considerations.

It is important to note that information contained by the notification packet may needed to be processed at multiple points in the router (e.g. multiple linecards may need to react on that message). This document describes the way of sending the information between nodes, but distributing this information inside the node (if needed) is out of the scope of this document.

3. Transport Logic - Distribution of the Notifications

The distribution of a notification to multiple receivers can be implemented in many ways. The main body of this draft describes some such options, however, other application specific distribution mechanisms may exist. Some more details can be found in the Appendix.

3.1. Flooding mode

In flooding mode, the IGP configures the dataplane cards to replicate each received FN message to each interface with a neighbour router in the same area.

This happens by making use of bidirectional multicast forwarding. In bidir multicast, all interfaces added to the multicast group can be incoming and outgoing interfaces as well. The principle is that a router replicates the incoming packet to *all* assigned interfaces except the incoming interface. If the local router is the source of the packet to be forwarded, then the packet is replicated to all interfaces. That is, the decision about which interfaces should actually be used as outgoing is determined on demand.

First, the FN service is assigned a multicast group address, let us call this MC-FN address. Then, the IGP assigns all interfaces to MC-FN which lead to neighbouring routers selected by the IGP.

When the FN service is instructed to disseminate a message, it creates an IP packet (as described below in Section 4) and sets its IP destination address to the MC-FN multicast address. This IP packet is then multicasted to all IGP neighbours in the area.

Recipients of FN multicast-forward the packet according to the rules of bidirectional multicast, i.e. to all interfaces which the local IGP pre-configured except the incoming interface. As this may cause loops without pre-caution (consider three routers in a triangle), before forwarding, therefore, the forwarding engine has to perform duplicate check.

3.1.1. Duplicate Check with Flooding

Duplicate check can be performed in numeruous ways.

Duplicate check can be performed by maintaining a short queue of previously forwarded FN messages. Before forwarding, if the FN message is found in the queue, then it was forwarded beforehand, so it may be dropped. Otherwise it should be forwarded and it should be added to the queue.

Alternatively, the queue may contain a signature of the previously forwarded FN messages, such as an MD5 or SHA256 signature or any other hash. This signature may be carried in the packet, e.g. due to authentication purposes, such as with the authentication mechanisms described in Section 4.2.1.

In either of the above queue-based mechanisms, the size of the queue can be set to a value that corresponds to the maximal number of legal FN messages generated by a single event. For instance, if FN is used to broadcast failure identifiers in case of failures, then it is likely that the failure of the node with the most neighbours will trigger the most FN messages (1 from each neighbour).

It is also possible to use application-dependent duplicate check: the state machine of the FN-application can be left responsible to decide whether the information carried in the packet contains new information or it is a duplicate. This is only useful in the case if the application can perform the duplicate check more efficiently than the above generic mechanisms. Presently, [I-D.csaszar-ipfrr-fn] specifies an application-specific duplicate check procedure.

3.2. Spanning Tree Mode

If reliable forwarding of notification packet is not always a strict requirement, spanning trees may be used for forwarding. In the simplest case, the nodes can build up a single spannig tree, and notification packets can be forwarded along this tree with bidirectional forwarding. This solution has the advantage that no duplicate check is needed. The tree may be built up with bidirectional PIM [RFC5015].

Another possibility is to use Maximally Redundant Trees [I-D.ietf-rtgwg-mrt-frr-architecture], a pair of spanning trees which give some failure tolerance. Since the common root of these trees can always be reached in the case of a single failure, and since the root can reach all the nodes, notification packets sent on both trees can tolerate any single failure, if the root propagates the packets it received on both trees. Further details about spanning trees are described in the Appendix.

4. Message Encoding

4.1. Seamless Encapsulation

An application may define its own message for FN to distribute quickly. In this case, only the special destination address (e.g. MC-FN) shows that the message was sent using the FN service.

In this case, the entire payload of the IP packet is determined by the application including sequence numbering and authentication. The IP packet’s protocol field can also be set by the application.

4.2. Dedicated FN Message

An alternative option is for the FN messages to be distributed in UDP datagrams with well-known port values in the UDP header that need to be allocated by IANA.

The FN packet format inside a UDP datagram is the following:

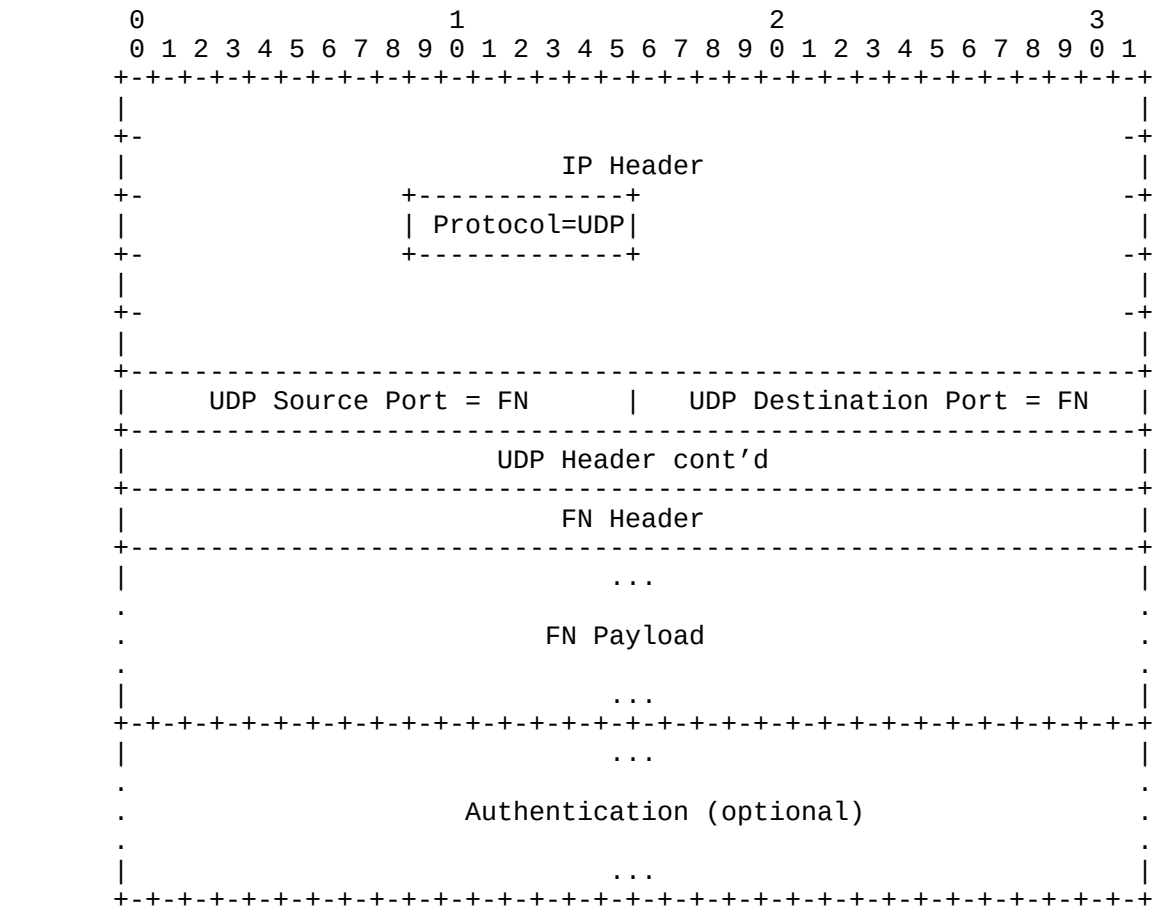


Figure 1: FN packet format as a UDP datagram

The encoding of the FN Header is as follows:

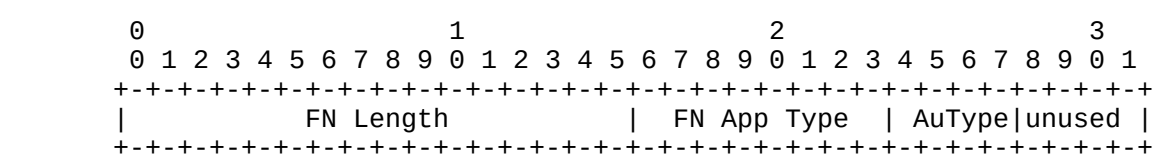


Figure 2: FN Header encoding

FN Length (16 bits)
The length of the FN message in bytes including the FN Header and the FN Payload. The authentication data optionally appended to the FN packet is not considered part of the FN message: the

authentication data is not included in the FN Length field, although it is included in the length field of the packet's IP header.

FN App Type (8 bits)
Identifies the application which should be the receiver of the notification. A value for each application needs to be assigned by IANA.

AuType
Identifies the authentication procedure to be used for the packet. Authentication options are discussed in Section 4.2.1 of the specification.

4.2.1. Authentication

Fast Notification intends to provide a trustable service option, so that receivers of FN packets are able to verify that the packet is sent by an authentic source. Simple password authentication and hash based authentication methods (with MD5 or SHA256) are described in the following subsections.

If AuType is set to 0x0, then the FN packet is not carrying an Authentication field at the end of the packet. Note that even in this case the FN application in the payload may still use its own authentication mechanism.

If AuType is non null, an Authentication field must be appended after the FN message. The encoding of this field is as described below.

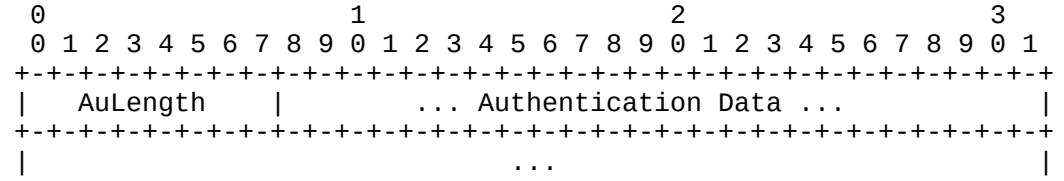


Figure 3: Authentication field in FN packets

AuLength
Describes the length of the entire Authentication field in bytes.

The authentication type may be manually pre-configured or may be selected automatically. For automatic selection, the nodes have to know what type of authentication is applicable for the rest of the nodes. This may achieved by extending the IGP to advertise the FN authentication capabilities. The most straightforward way to achieve

this is to extend the Router Capability TLVs available both in OSPF [RFC4970] and in IS-IS [RFC4971].

4.2.1.1. Area-scoped and Link-scoped Authentication

Since FN is a solution to disseminate an event notification from one source to a whole area of nodes, the simplest approach would be to use per-area authentication, e.g., a common password, a common pre-shared key among all nodes in the area as described in the following sub-sections, or digital signatures.

Carriers may, however, prefer per-link authentication. In order not to lose the speed (simple per-hop processing, fast forwarding property) of FN, link-scoped authentication is suggested only if the forwarding plane supports it, i.e. if there is hardware support to verify and re-generate authentication hop-by-hop. In such cases, the operator may need to configure a common pre-shared key only on routers connected by the same link. It is even possible that there is no authentication on some links considered safe.

4.2.1.2. Simple Password Authentication

Simple password authentication guards against routers inadvertently joining the routing area; each router must first be configured with a password before it can participate in Fast Notification.

The password is stored in the Authentication Data field. AuLength is set to the length of the password in bytes plus 1. Two AuType values for simple password authentication need to be allocated by IANA: one for area-scope and another for link-scoped.

With per-link authentication mode, the Authentication field must be stripped and regenerated hop-by-hop.

Simple password authentication, however, can be easily compromised as anyone with physical access to the network can read the password.

4.2.1.3. Cryptographic Authentication for FN

Using this authentication type, a secret key is used to generate/ verify a "message digest" that is appended to the end of the FN packet. The message digest is a one-way function of the FN packet and the secret key. This authentication mechanism resembles the cryptographic authentication mechanism of [RFC2328].

4.2.1.3.1. MD5

The packet signature is created by an MD5 hash performed on an object which is the concatenation of the FN message, including the FN header, and the pre-shared secret key. The resulting 16 byte MD5 message digest is appended to the FN message into the Authentication field as shown below.

The AuType in the FN header is set to indicate cryptographic authentication, the specific value is to be assigned by IANA both for area-scoped and for link-scoped versions.

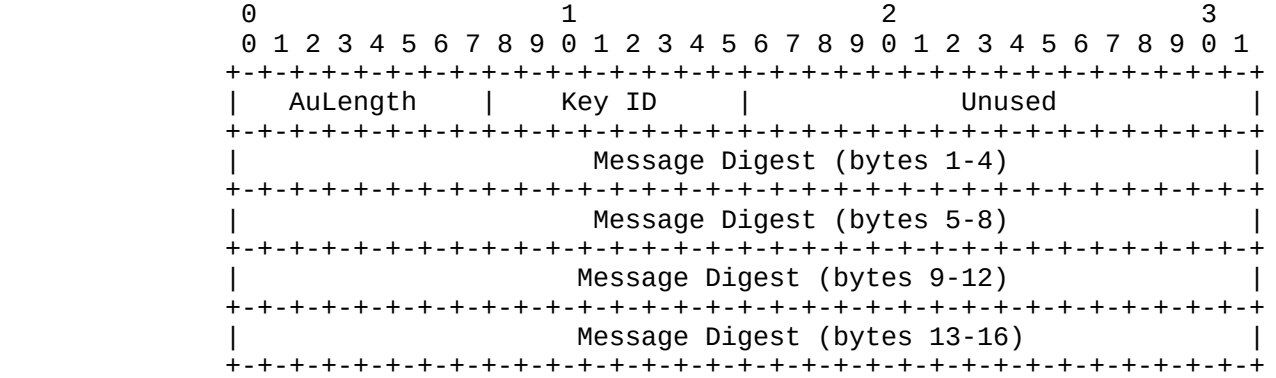


Figure 4: Authentication field in FN packets with MD5 cryptographic authentication.

AuLength
AuLength is set to 20 bytes.

Key ID
This field identifies the algorithm and secret key used to create the message digest appended to the FN packet. This field allows that multiple pre-shared keys may exist in parallel.

Message Digest
The 16 byte long MD5 hash performed on an object which is the concatenation of the FN message, including the FN header, and the pre-shared secret key identified by Key ID.

When receiving an FN message, if the FN header indicates MD5 authentication, then the last 20 bytes of the FN message are set aside. The recipient forwarding plane element calculates a new MD5 digest of the remainder of the FN message to which it appends its own known secret key identified by Key ID. The calculated and received digests are compared. In case of mismatch, the FN message is discarded.

In per-link authentication mode, the Authentication field must be regenerated hop-by-hop using the key of the outgoing link.

4.2.1.3.2. SHA256

Similarly to how MD5 authentication works, it is possible to use Secure Hash 256 hash. Currently this is a more secure hash function than MD5. The Authentication field would look like this:

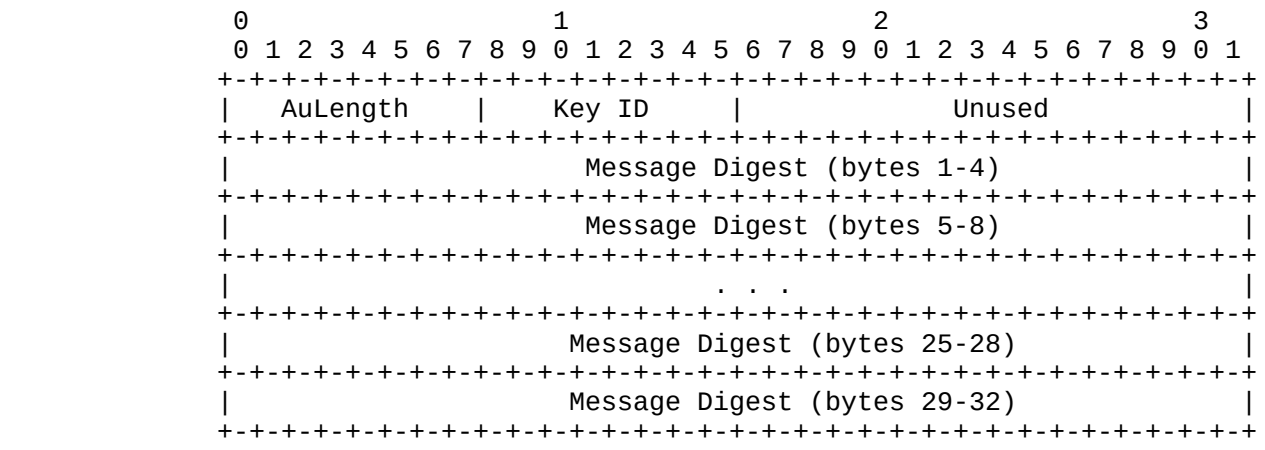


Figure 5: Authentication field in FN packets with MD5 cryptographic authentication.

AuLength
AuLength is set to 36 bytes.

Key ID
This field identifies the algorithm and secret key used to create the message digest appended to the FN packet. This field allows that multiple pre-shared keys may exist in parallel.

Message Digest
The 32 bytes long SHA256 value calculated on an object which is the concatenation of the FN message, including the FN header, and the pre-shared secret key identified by Key ID.

When receiving an FN message, if the FN header indicates SHA256 authentication, then the last 68 bytes of the FN message are set aside. The recipient forwarding plane element calculates a new SHA256 digest of the remainder of the FN message to which it appends its own known secret key identified by Key ID. The calculated and received digests are compared. In case of mismatch, the FN message is discarded.

In per-link authentication mode, the Authentication field must be regenerated hop-by-hop using the key of the outgoing link.

4.2.1.3.3. Digital Signatures

A router may choose to use public key cryptography to digitally sign the notification to provide certification of authenticity. This mechanism can avoid shared secret that is required for other authentication mechanisms described in this document. This authentication mechanism resembles the authentication mechanism of OSPF with digital signatures as defined in [RFC2154].

5. Security Considerations

This draft has described basic optional procedures for authentication. The mechanism, however, does not protect against replay attacks.

If an application of FN require protection against replay attacks, then these applications should provide their own specific sequence numbering within the FN payload. Recipient applications should accept FN messages only if the included sequence number is valid.

Since the message digest of cryptographic authentication also covers the payload, even if an attacker knew how to construct the new sequence number, it would not be able to generate a correct message digest without the pre shared key. This way, a sequence number in the payload combined with FN’s cryptographic authentication offers sufficient protection against replay attacks.

6. FN Packet Processing Summary

When receiving an FN packet, a node has to perform the following steps.

It has to identify that the packet is an FN packet. This can be done utilising the destination IP address (MC-FN) or by inspecting the UDP port field.

If the flooding like transport logic described in Section 3 is used the node has to perform duplicate check following the teachings in Section 3.1.1.

If AuType is non-null, the node has to perform authentication check as discussed in Section 4.2.1.

To protect against replay attacks, the node shall perform verification of the sequence number provided by the application.

Punt and forward. The notification may need to be multicasted but it also needs to be punted to the local application on the linecard to start processing.

Authentication check, sequence number check and punting/forwarding may commence in any order deemed necessary by the operator. If the operator prefers highest level of security, then both checks should be performed before forwarding. If, however, the operator prefers per-hop performance but still wants to ensure that malice packets cannot harm the network, then authentication and sequence number checks may also happen after punting the packet, i.e. before processing the information contained inside the FN payload. In this case, malicious packets may get propagated to every node but they still do not cause any change in the configuration.

7. IANA Considerations

A UDP port value needs to be assigned by IANA for FN. IANA also needs to maintain values for FN App Type as applications are being proposed.

Multicast addresses used for the distribution trees are either allocated by IANA or they can be a configuration parameter within the local domain.

8. Acknowledgements

The authors owe thanks to Acee Lindem, Joel Halpern and Jakob Heitz for their review and comments. Also thanks to Alia Atlas for constructive feedback.

9. References

9.1. Normative References

[I-D.enyedi-rtgwg-mrt-frr-algorithm]
Envedi, G., Csaszar, A., Atlas, A., cbowers@juniper.net, c., and A. Gopalan, "Algorithms for computing Maximally Redundant Trees for IP/LDP Fast- Reroute", draft-enyedi-rtgwg-mrt-frr-algorithm-03 (work in progress), July 2013.

[I-D.ietf-rtgwg-mrt-frr-architecture]
Atlas, A., Kebler, R., Envedi, G., Csaszar, A., Tantsura, J., Konstantynowicz, M., and R. White, "An Architecture for IP/LDP Fast-Reroute Using Maximally Redundant Trees", draft-ietf-rtgwg-mrt-frr-architecture-03 (work in progress), July 2013.

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

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

[RFC4970] Lindem, A., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 4970, July 2007.

[RFC4971] Vasseur, JP., Shen, N., and R. Aggarwal, "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, July 2007.

[RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, October 2007.

9.2. Informative References

[Eny2009] Enyedi, G., Retvari, G., and A. Csaszar, "On Finding Maximally Redundant Trees in Strictly Linear Time, IEEE Symposium on Computers and Communications (ISCC)", 2009.

[I-D.csaszar-ipfrr-fn] Csaszar, A., Envedi, G., Tantsura, J., Kini, S., Sucec, J., and S. Das, "IP Fast Re-Route with Fast Notification", draft-csaszar-ipfrr-fn-03 (work in progress), June 2012.

[RFC2154] Murphy, S., Badger, M., and B. Wellington, "OSPF with Digital Signatures", RFC 2154, June 1997.

Appendix A. Further Options for Transport Logic

The options described in this appendix represent alternative solutions to the flooding based approach described in Section Section 3.

It is left for WG discussion and further evaluation to decide whether any of these options should potentially be preferred instead of redundant trees.

A.1. Multicast Tree-based Transport

One way of transporting an identical piece of information to several receivers at the same time is to use multicast distribution trees. A tree based transport solution is beneficial since multicast support is already implemented in all forwarding entities, so it is possible to use existing implementations.

With multicast or tree based transport, the Fast Notification (FN) packet can be recognized by a pre-configured or well known destination IP address, denoted by MC-FN in the following, which is the group address of the FN service.

If the FN service is triggered to send out a notification, the notification will be encapsulated in a new IP packet, where the destination IP address is set to MC-FN.

A.1.1. Fault Tolerance of a Single Distribution Tree

Several solutions described in this draft use a single tree to disseminate a notification from one given source.

The single tree solution is simple, however it is not redundant: a single failure may partition the tree, which will prevent notifications from reaching some nodes in the area.

Different applications may have different needs for reliability. For example, when we use fast notification to disseminate network failure information, all nodes surrounding the failure can detect and originate the failure notifications independently. Any one of these notifications (or a subset of them) may be sufficient for the application to make the right decision. This draft provides several different transport options from which an applications can choose.

A.1.2. Pair of Redundant Trees

If an FN application needs the exact same data to be distributed in the case of any single node or any single link failure, the FN service could opt to run in "redundant tree mode".

A pair of "maximally redundant trees" [I-D.enyedi-rtgwg-mrt-frr-algorithm] ensures that at each single node or link failure each node still reaches the common root of the trees through at least one of the trees. A redundant tree pair is a known prior-art graph-theoretical object that is possible to find on any 2-node connected network. Even better, it is even possible to find maximally redundant trees in networks where the 2-node connected criterion does not "fully" hold (e.g. there are a few cut vertices) [Eny2009], [I-D.ietf-rtgwg-mrt-frr-architecture].

Note that the referenced algorithm(s) build a pair of trees considering a specific root. The root can be selected in different ways, the only thing that is important that each node makes the same selection, consistently. For instance, the node with the highest or lowest router ID can be used.

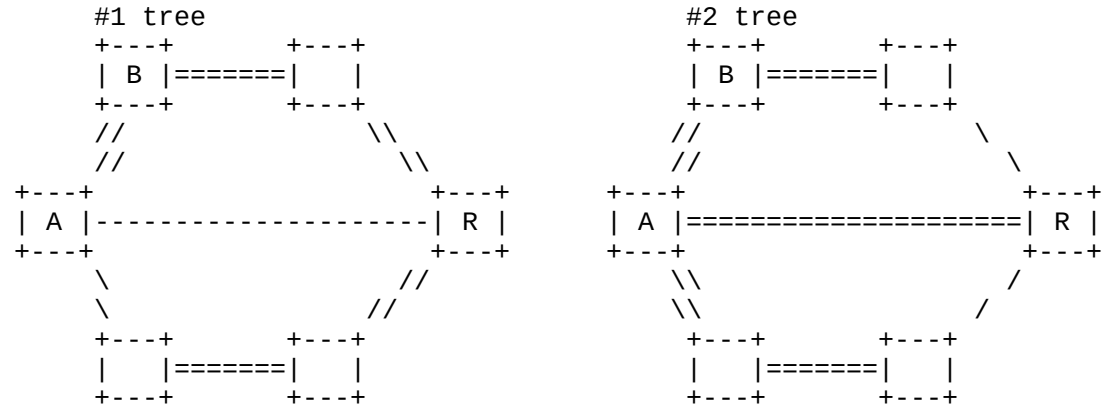


Figure 6: Example: a pair of redundant trees (double lines) of a common root R

There is one special constraint in building the redundant trees. A (maximally) redundant tree pair is needed, where in one of the trees the root has only one child in order to protect against the failure of the root itself. Algorithms presented in [Eny2009], [I-D.enyedi-rtgwg-mrt-frr-algorithm] produce such trees.

In redundant-tree mode, each node multicasts the requested notification on both trees, if it is possible, but at least along one of the trees. Redundant trees require two multicast group addresses. MC-FN identifies one of the trees, and MC-FN-2 identifies the other tree.

Each node multicast forwards the received notification packet (on the same tree). The root node performs as every other node but in addition it also multicast the notification on the other tree! I.e. it forwards a replica of the incoming notification in which it replaces the destination address identifying the other multicast distribution tree.

When the network remains connected and the root remains operable after a single failure, the root will be reached on at least one of the trees. Thus, since the root can reach every node along at least one of the trees, all the notifications will reach each node. However, when the root or the link to the root fails, that tree, in which the root has only one child, remains connected (the root is a leaf there), thus, all the nodes can be reached along that tree.

For example, let us consider that in Figure 6 FN is used to disseminate failure information. If link A-B fails, the notifications originating from node B (e.g. reporting that the

connectivity from B to A is lost) will reach R on tree #1. Notifications originating from A (e.g. reporting that the connectivity from A to B is lost) will reach R on tree #2. From R, each node is reachable through one of the trees, so each node will be notified about both events.

A.2. Unicast

This method addresses the need in a unique way. It has the following properties:

Plain simple, without the need of any forwarding plane change or cooperation;

Short turnaround time (i.e. ready for next hit);

100% link break coverage (may not work in certain node failure cases);

Little change to OSPF (need encapsulation for IS-IS).

A.2.1. Method

The method is simple in design, easy to implement and quick to deploy. It requires no topology changes or specific configurations. It adds little overhead to the overall system.

The method sends the event message to every router in the area in an IP packet. This appears burdensome to the sending router which has to duplicate the packet sending effort many times. Practical experience has shown, however, that the amount of effort is not a big concern in reasonable sized networks.

Normal flooding (regular or fast) process requires a router to duplicate the packet to all flooding eligible interfaces. All routers have to be fast-flooding-aware. This implies new code to every router in control plane and/or forwarding plane.

The method uses a different approach. It takes advantage of the given routing/forwarding table in each router in the IP domain. The originating router of the flooding information simply sends multiple copies of the packet to each and every router in the domain. These packets are forwarded to the destination routers at forwarding plane speed,

just like the way the regular IP data traffic is handled. No special handling in any other routers is needed.

This small delay on the sender can be minimized by pre-downloading the link-broken message packets to the forwarding plane. Since the forwarding plane already has the list of all routers which are part of the IGP routing table, the forwarding plane can dispatch the packet directly.

In essence, the flooding in this method is tree based, just like a multicast tree. The key is that no special tree is generated for this purpose; the normal routing table which is an SPF tree (SPT) plays a role of the flooding tree. This logic guarantees that the flooding follows the shortest path and no flooding loop is created.

A.2.2. Sample Operation

Figure 7 depicts a scenario where router A wants to flood its message to all other routers in the domain using the unicast flooding method.

Instead of sending one packet to each of its neighbor, and letting the neighbor flood the packet further, router A directly send the same packet to each router in the domain, one at a time. In this sample network, router A sends out 5 packets.

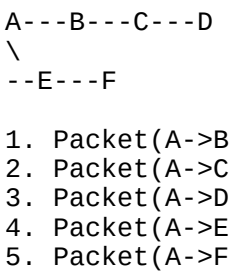


Figure 7: Multiple Unicast Packets

The unicast flooding procedure is solely controlled by the sending router. No action is needed from other routers other than their normal forwarding functionalities. This method is extremely simple and useful for quick prototyping and deployment.

A.3. Gated Multicast through RPF Check

This method fulfills the purpose with the following characters:

1. No need to build the multicast tree. It is the same as the SPT computed by the IGP routing process;

2. Flooding loops are prevented by RPF Check.

The method has all the benefits of multicast flooding. It, however, does not require running multicast protocol to setup the multicast tree. The unicast shortest path tree is used as a multicast tree.

A.3.1. Loop Prevention - RPF Check

In this mechanism, the distribution tree is not explicitly built. Rather, each node will first do a Reverse Path Forwarding (RPF) check before it floods the notification to other links.

A special multicast address is defined and is subject to IANA approval. This address is used to qualify the notification packet for fast flooding. When a notification packet arrives, the receiving node will perform an IP unicast routing table lookup for the originator IP address of the notification and find the outgoing interface. Only when the arriving interface of the notification is the same as the outgoing interface leading towards the originator IP address, will the notification be flooded to other interfaces.

IP Multicast forwarding with RPF check is available on most of the routing/switching platforms. To support flooding with RPF check, a special IP multicast group must be used. A bi-directional IP multicast forwarding entry is created that consists of all interfaces within the flooding scope, typically an IGP area.

A.3.2. Operation

The Gated flooding operation is illustrated in Figure 8.

```
All Routers, IGP Process:
if (SPT ready) {
 duplicate the SPT as Bidir_Multicast_tree;
 download the multicast_tree to forwarding plane;
}
add FNF_multicast_group_addr;

    Sender of the FNF notification:
if (breakage detected) {
 pack the notification in a packet;
 send the packet to the FNF_multicast_group_addr;
}

Receiver of the FNF notification:
 if (notification received) {
 if (RPC_interface == incoming_interface) {
  multicast the notification to all other interfaces;
 }
 forward the notification to IGP for processing;
```

}

Figure 8: Gated flooding operation

Figure 9 shows a sample operation on a four-router mesh network. The left figure is the topology. The right figure is the shortest path tree rooted at A.

Router A initiates the flooding. But the downstream routers B, C, and D will drop all messages except the ones that come from their shortest path parent node. For example, A's message to C via B is dropped by C, because C knows that its reverse path forwarding (RPF) nexthop is A.

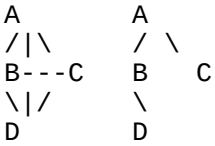


Figure 9: Loop Prevention through the RPF check

A.4. Further Multicast Tree based Transport Options

A.4.1. Source Specific Trees

One implementation option is to rely on source specific multicast. This means that even though there is only a single multicast group address (MC-FN) allocated to the FN service, the FIB of each router is configured with forwarding information for as many trees as many FN sources (nodes) there are in the routing area, i.e. to each (S_i ,MC-FN) pair.

A.4.2. A Single Bidirectional Shared Tree

In the previous solution each source specific tree is a spanning tree. It is possible to reduce the complexity of managing and configuring n spanning trees in the area by using bidirectional shared trees. By building a bidirectional shared tree, all nodes on the tree can send and receive traffic using that single tree. Each sent packet from any source is multicasted on the tree to all other receivers.

The tree must be consistently computed at all routers. For this, the following rules may be given:

The tree can be computed as a shortest path tree rooted at e.g. the highest router-id. When multiple paths are available, the

neighbouring node in the graph e.g. with highest router-id can be picked. When multiple paths are available through multiple interfaces to a neighbouring node, e.g. a numbered interface may be preferred over an unnumbered interface. A higher IP address may be preferred among numbered interfaces and a higher ifIndex may be preferred among unnumbered interfaces.

Note, however, that the important point is that the rules are consistent among nodes. That is, a router may pick the lower router IDs if it is ensured that ALL routers will do the same to ensure consistency.

Multicast forwarding state is installed using such a tree as a bi-directional tree. Each router on the tree can send packets to all other routers on that tree.

Note that the multicast spanning tree can be built using [RFC5015] so that each router within an area subscribes to the same multicast group address. Using BIDIR-PIM in such a way will eventually build a multicast spanning tree among all routers within the area. (BIDIR-PIM is normally used to build a shared, bidirectional multicast tree among multiple sources and receivers.)

A.5. Layer 2 Networks

Layer 2 (e.g. Ethernet) networks offer further options for distributing the notification (e.g. using spanning trees offered by STP). Definition of these is being considered and will be included in a future revision of this draft.

Authors' Addresses

Wenhu Lu
Ericsson
300 Holger Way
San Jose, California 95134
USA

Email: Wenhu.Lu@ericsson.com

Sriganesh Kini
Ericsson
300 Holger Way
San Jose, California 95134
USA

Email: Sriganesh.Kini@ericsson.com

Andras Csaszar (editor)
Ericsson
Irinyi J utca 4-10
Budapest 1117
Hungary

Email: Andras.Csaszar@ericsson.com

Gabor Sandor Enyedi
Ericsson
Irinyi J utca 4-10
Budapest 1117
Hungary

Email: Gabor.Sandor.Enyedi@ericsson.com

Jeff Tantsura
Ericsson
300 Holger Way
San Jose, California 95134
USA

Email: Jeff.Tantsura@ericsson.com

Network Working Group
Internet Draft
Intended Status: Informational
Expires: August 2011

N. So
A. Malis
D. McDysan
Verizon
L. Yong
Huawei
F. Jounay
France Telecom
Y. Kamite
NTT

February 22, 2011

Composite Link Framework in Multi Protocol Label Switching (MPLS)

draft-so-yong-rtgwg-cl-framework-03

at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on August 22, 2011.

Abstract

This document specifies a composite link framework in MPLS network. A composite link consists of a group of homogenous or non-homogenous links that have the same forward adjacency and can be considered as a single TE link or an IP link in routing. The composite link relies on its component links to carry the traffic over composite link. The document specifies composite link model. Applicability is described for a single pair of MPLS-capable nodes, a sequence of MPLS-capable nodes, or a set of layer networks connecting MPLS-capable nodes.

Table of Contents

1. Introduction

Composite link functional requirements are specified in [CL-REQ]. This document specifies a framework of Composite Link in MPLS network to meet the requirements. Single link and link bundle [RFC4201] have been widely used in today's MPLS networks. A link bundle bundles a group of homogeneous links as a TE link to make routing approach more scalable. A composite link allows bundling non-homogenous links together as a single logical link. The motivations for using a composite link are descried in the document [CL-REQ]. This document describes composite link framework in the context of MPLS network with MPLS control plane.

A composite link is a single logical link in MPLS network that contains multiple parallel component links between two routers. Unlike a link bundle [RFC4201], the component links in a composite link can have different properties such as cost or capacity. A composite link can transport aggregated traffic as other physical links from the network perspective and use its component links to carry the traffic internally.

Specific protocol solutions are outside the scope of this document.

2. Conventions used in this document

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

2.1. Terminology

Composite Link: A composite link is a logical link composed of a set of parallel point-to-point component links, where all links in the set share the same endpoints. A composite link may itself be a component of another composite link, but only a strict hierarchy of links is allowed.

Component Link: A point-to-point physical or logical link that preserves ordering in the steady state. A component link may have transient out of order events, but such events must not exceed the network's specific NPO. Examples of a physical link are: Lambda, Ethernet PHY, and OTN. Examples of a logical link are: MPLS LSP, Ethernet VLAN, and MPLS-TP LSP.

Flow: A sequence of packets that must be transferred in order on one component link.

Flow identification: The label stack and other information that uniquely identifies a flow. Other information in flow identification may include an IP header, PW control word, Ethernet MAC address, etc. Note that an LSP may contain one or more Flows or an LSP may be equivalent to a Flow. Flow identification is used to locally select a component link, or a path through the network toward the destination.

Network Performance Objective (NPO): Numerical values for performance measures, principally availability, latency, and delay variation. See Appendix A for more details.

3. Composite Link Framework

A Composite Link in the context of MPLS network is a set of parallel links between two routers that form a single logical link within the network. Composite link model is illustrated in Figure 1, where a composite link is configured between routers R1 and R2. The composite link has three component links. Individual component links in a composite link may be supported by different transport technologies such as wavelength, Ethernet VLAN. Even if the transport technology implementing the component links is identical, the characteristics (e.g., bandwidth, latency) of the component links may differ.

As shown in Figure 1, the composite link may carry LSP traffic flows and control plane packets that appear as IP packets. A LSP may be established over the link by either RSVP-TE or LDP signaling protocols. All component links in a composite link have the same forwarding adjacency. The composite link forms one routing interface at the composite link end points for MPLS control plane. In other words, two routers connected via a composite link have forwarding adjacency and routing adjacency. Each component link only has significance to the composite link, i.e. it does not appear as a link in the control plane.

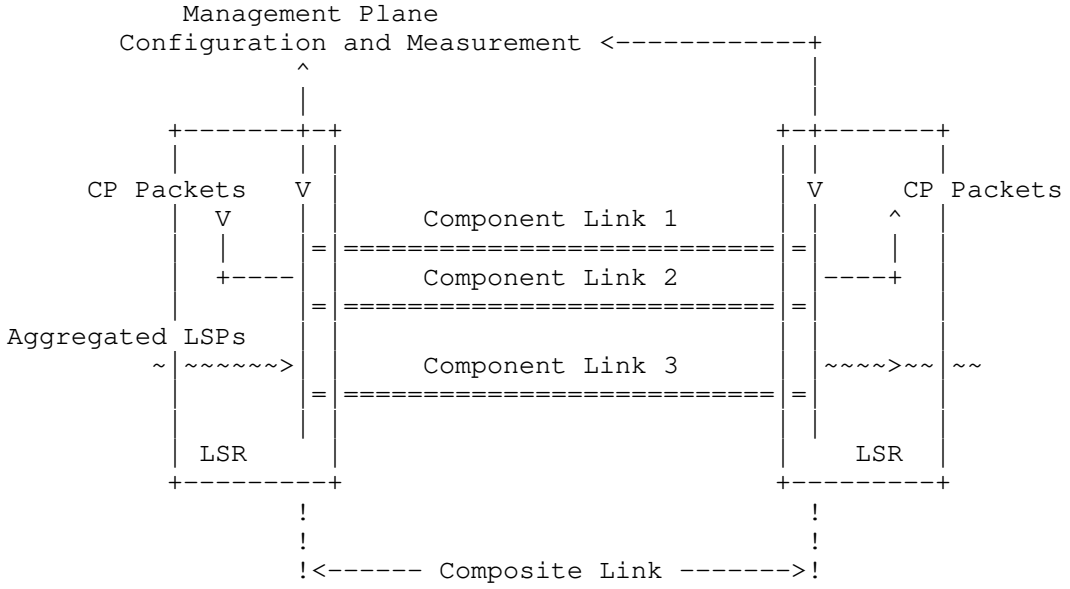


Figure 1 Composite Link Architecture Model

A component link in a composite link may be constructed in different ways.[CL-REQ] Figure 2 shows three common ways that may be deployed in a network.

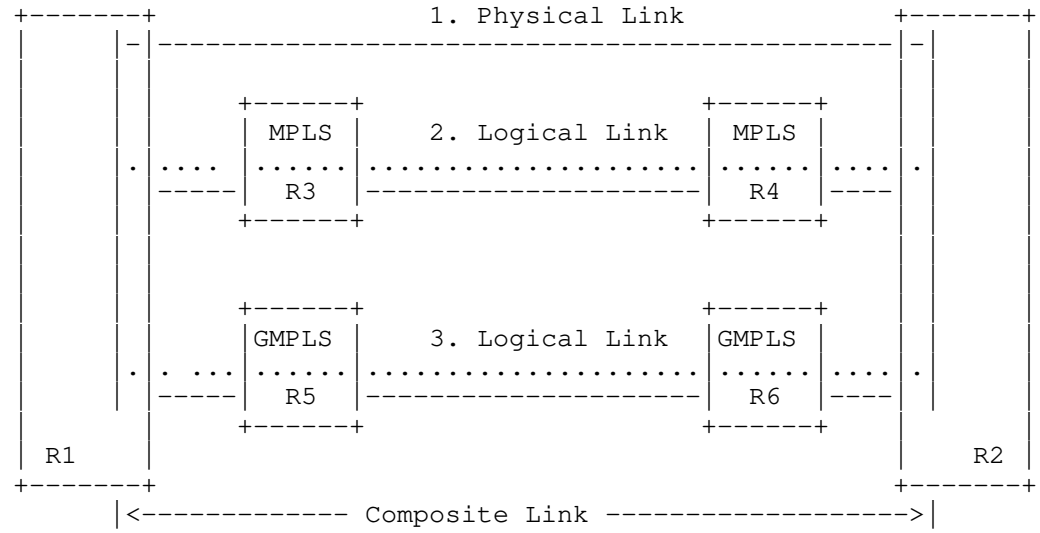


Figure 2 Illustration of Component Link Variances

As shown, the first component link is configured with direct physical media wire. The second component link is a TE tunnel that traverses R3 and R4. Both R3 and R4 are the nodes in the MPLS. The third component link is formed by lower layer network that has GMPLS enabled. In this case, R5 and R6 are not the nodes controlled by the MPLS but provide the connectivity for the component link. Note: if two unidirectional LSPs are used to construct a component link, they MUST be co-routed.

Composite link forms one logical link between connected routers and is used to carry aggregated traffic.[CL-REQ] Composite link relies on its component links to carry the traffic over the composite link. This means that a composite link maps incoming traffic into component links. The router (R1 in Figure 1) of composite link ingress maps a set of traffic flows including control plane packets to a specific component link. The router (R2 in Figure 1) of composite link egress receives the packets from its component links and sends them to MPLS forwarding engine like a regular link. The traffic from R2 to R1 is distributed by the router R2.

Traffic mapping to component links may be done by control plane, management plane, or data plane.[CL-REQ] The objective is to keep the individual flow packets in sequence and do not overload any component link.[CL-REQ] Operator may have other objectives such as place a bi-directional flow or LSP on the same component link in both direction, load balance over component links, composite link energy saving, and etc. A flow may be a LSP, or sub-LSP [MLSP], PW, a flow within PW [FAT-PW], entropy flow in LSP [Entropy].

4. Composite Link in Control Plane

A composite Link is advertised as a single logical interface between two connected routers, which forms routing and forwarding adjacency between the routers in IGP. The interface parameters for the composite link can be pre-configured by operator or be derived from its component links. Composite link advertisement requirements are specified in [CL-REQ].

In IGP-TE, a composite link is advertised as a single TE link between two connected routers. This is similar to a link bundle [RFC4201]. Link bundle applies to a set of homogenous component links. Composite link allows homogenous and non-homogenous component links. The link bundle protocol extension for composite link advertisement is for further study.

A composite link may contain the set of component links. A component link may be configured by operator or signaled by the control plane. If two unidirectional LSPs are used to construct a component link, they MUST be co-routed. In both cases, it is necessary to convey component link parameters to the composite link.[CL-REQ]

When a component link is supported by lower layer network (third component link in figure 2), the control plane that the composite link resides is able to interoperate with the GMPLS or MPLS-TP control plane that lower layer network uses for component link addition and deletion.[CL-REQ]

It is possible for operator to configure one or multiple interface (s) over a composite link.

Both LDP [RFC5036] and RSVP-TE [RFC3209] can be used to signal a LSP over a composite link. The router of composite link ingress MUST place the LSP on the component link that meets the LSP criteria indicated in the signal message.

Since composite link capacity is aggregated capacity and is often larger than individual component link capacity, it is possible to signal a LSP whose BW is larger than individual component link capacity.[CL-REQ] Assumption is such LSP carrying an aggregated traffic.

When a bi-directional LSP request is signaled over a composite link, if the request indicates that the LSP must be placed on the same component link, the routers of the composite link MUST place the LSP traffic in both directions on a same component link.

5. Composite Link in Data Plane

The traffic over a composite link is distributed over individual component links. Traffic dissemination may be determined by control plane, management plane, or data plane, and may be changed due to component link status change.[CL-REQ] The distribution function is local to the routers in which a composite link belongs to and is not specified here. However, if a bi-directional LSP is required to be placed on the same component link in both directions, the routers at both composite link end points need cooperation in determining the component link for the LSP. The protocol extension of that is for further study.

A component link in a composite link may fail independently. The routers of a composite link are able to recognize component link failure and re-assign impacted flows to other active component links in minimal disruptive manner. When a composite link is not able to transport all flows, it preempts some flows based upon local management configuration and informs the control plane on these preempted flows. This action ensures the remaining traffic is transported properly.

The composite link functions provide component link fault notification and composite link fault notification. Component link fault notification is sent to the management plane. Composite link fault notification is sent to the control plane and management plane.

Operator may want to perform an optimization function such as load balance or energy saving over a composite link, which may conduct some traffic moving from one component link to another. The process MUST support locally and gracefully traffic movement process among component links. The protocol that facilitates this process between two composite link end points is for further study.

6. Composite Link in Management Plane

Management Plane MUST keep tracking a composite link and its individual composite link status and configuration. Management Plane MUST be able to make any component link in a composite link active and de-activate in order to facilitate operation maintenance task. The routers of a composite link resides MUST perform the redistribution of the traffic flows on a de-activated link to other component links based on the traffic flow TE criteria.

Management Plane MUST be able to configure a LSP over a composite link and be able to select a component link for the LSP.

Management Plane MUST be able to trace which component link a LSP is assigned to and monitor individual component link and composite link performance.

Management Plane MUST be able to ping individual component link within a composite link.

Management Plane should build the proper commands to allow operator execute an optimization process.

7. Security Considerations

For further study.

8. IANA Considerations

IANA actions to provide solutions are for further study.

9. References

9.1. Normative References

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

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

[RFC4201] Kompella, K., "Link Bundle in MPLS Traffic Engineering", RFC 4201, March 2005.

[RFC5036] Andersson, L., "LDP Specification", RFC 5036 , October 2007.

9.2. Informative References

[CL-REQ] Villamizar, C. and McDysan, D, "Requirements for MPLS Over Composite Link", Oct. 2010, Work in Progress

[Entropy] Kompella, K. and S. Amante, "The Use of Entropy Labels in MPLS Forwarding", draft-ietf-kompella-mpls-entropy-label-01.txt, November 2008, Work in Progress

[FAT-PW] Bryan, S., et. Al, "Flow Aware Transport of Pseudowire over an MPLS PSN", draft-ietf-pwe3-fat-pw-05, Feb. 2011, Work in progress

[MLSP] Kompella, K. "Multi-path Label Switched Paths Signaled Using RSVP-TE", draft-kompella-mpls-rsvp-ecmp-00.txt, July 2010, Work in Progress

10. Acknowledgments

Authors would like to thank Adrian Farrel for his extensive comments and suggestions, Ron Bonica, Nabil Bitar, Eric Gray, Lou Berger, and Kireeti Kompella for their reviews and great suggestions.

Authors' Addresses

So Ning
Verizon
2400 N. Glem Ave.,
Richerdson, TX 75082
Phone: +1 972-729-7905
Email: ning.so@verizonbusiness.com

Andrew Malis
Verizon
117 West St.
Waltham, MA 02451
Phone: +1 781-466-2362
Email: andrew.g.malis@verizon.com

Dave McDysan
Verizon
22001 Loudoun County PKWY
Ashburn, VA 20147
Email: dave.mcdysan@verizon.com

Lucy Yong
Huawei USA
1700 Alma Dr. Suite 500
Plano, TX 75075
Phone: +1 469-229-5387
Email: lucyyong@huawei.com

Frederic Jounay
France Telecom
2, avenue Pierre-Marzin
22307 Lannion Cedex,
FRANCE
Email: frederic.jounay@orange-ftgroup.com

Yuji Kamite
NTT Communications Corporation
Granpark Tower
3-4-1 Shibaura, Minato-ku
Tokyo 108-8118
Japan
Email: y.kamite@ntt.com

RTGWG
Internet-Draft
Intended status: Informational
Expires: December 31, 2012

S. Ning
Tata Communications
D. McDysan
Verizon
E. Osborne
Cisco
L. Yong
Huawei USA
C. Villamizar
Outer Cape Cod Network
Consulting

June 29, 2012

Composite Link Framework in Multi Protocol Label Switching (MPLS)
draft-so-yong-rtgwg-cl-framework-06

Abstract

This document specifies a framework for support of composite link in MPLS networks. A composite link consists of a group of homogenous or non-homogenous links that have the same forward adjacency and can be considered as a single TE link or an IP link in routing. A composite link relies on its component links to carry the traffic over the composite link. Applicability is described for a single pair of MPLS-capable nodes, a sequence of MPLS-capable nodes, or a set of layer networks connecting MPLS-capable nodes.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 31, 2012.

Copyright Notice

Table of Contents

1. Introduction

Composite Link functional requirements are specified in [I-D.ietf-rtgwg-cl-requirement]. Composite Link use cases are described in [I-D.symmvo-rtgwg-cl-use-cases]. This document specifies a framework to meet these requirements.

Classic multipath, including Ethernet Link Aggregation has been widely used in today's MPLS networks [RFC4385][RFC4928]. Classic multipath using non-Ethernet links are often advertised using MPLS Link bundling. A link bundle [RFC4201] bundles a group of homogeneous links as a TE link to make IGP-TE information exchange and RSVP-TE signaling more scalable. A composite link allows bundling non-homogenous links together as a single logical link. The motivations for using a composite link are descried in [I-D.ietf-rtgwg-cl-requirement] and [I-D.symmvo-rtgwg-cl-use-cases].

This document describes a composite link framework in the context of MPLS networks using an IGP-TE and RSVP-TE MPLS control plane with GMPLS extensions [RFC3209][RFC3630][RFC3945][RFC5305].

A composite link is a single logical link in MPLS network that contains multiple parallel component links between two MPLS LSR. Unlike a link bundle [RFC4201], the component links in a composite link can have different properties such as cost or capacity.

Specific protocol solutions are outside the scope of this document, however a framework for the extension of existing protocols is provided. Backwards compatibility is best achieved by extending existing protocols where practical rather than inventing new protocols. The focus is on examining where existing protocol mechanisms fall short with respect to [I-D.ietf-rtgwg-cl-requirement] and on extensions that will be required to accommodate functionality that is called for in [I-D.ietf-rtgwg-cl-requirement].

1.1. Architecture Summary

Networks aggregate information, both in the control plane and in the data plane, as a means to achieve scalability. A tradeoff exists between the needs of scalability and the needs to identify differing path and link characteristics and differing requirements among flows contained within further aggregated traffic flows. These tradeoffs are discussed in detail in Section 3.

Some aspects of Composite Link requirements present challenges for which multiple solutions may exist. In Section 4 various challenges and potential approaches are discussed.

A subset of the functionality called for in [I-D.ietf-rtgwg-cl-requirement] is available through MPLS Link Bundling [RFC4201]. Link bundling and other existing standards applicable to Composite Link are covered in Section 5.

The most straightforward means of supporting Composite Link requirements is to extend MPLS protocols and protocol semantics and in particular to extend link bundling. Extensions which have already been proposed in other documents which are applicable to Composite Link are discussed in Section 6.

Goals of most new protocol work within IETF is to reuse existing protocol encapsulations and mechanisms where they meet requirements and extend existing mechanisms such that additional complexity is minimized while meeting requirements and such that backwards compatibility is preserved to the extent it is practical to do so. These goals are considered in proposing a framework for further protocol extensions and mechanisms in Section 7.

1.2. Conventions used in this document

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

1.2.1. Terminology

Terminology defined in [I-D.ietf-rtgwg-cl-requirement] is used in this document.

The abbreviation IGP-TE is used as a shorthand indicating either OSPF-TE [RFC3630] or ISIS-TE [RFC5305].

2. Composite Link Key Characteristics

[I-D.ietf-rtgwg-cl-requirement] defines external behavior of Composite Links. The overall framework approach involves extending existing protocols in a backwards compatible manner and reusing ongoing work elsewhere in IETF where applicable, defining new protocols or semantics only where necessary. Given the requirements, and this approach of extending MPLS, Composite Link key characteristics can be described in greater detail than given requirements alone.

2.1. Flow Identification

Traffic mapping to component links is a data plane operation. Control over how the mapping is done may be directly dictated or constrained by the control plane or by the management plane. When unconstrained by the control plane or management plane, distribution of traffic is entirely a local matter. Regardless of constraints or lack or constraints, the traffic distribution is required to keep packets belonging to individual flows in sequence and meet QoS criteria specified per LSP by either signaling or management [RFC2475][RFC3260]. A key objective of the traffic distribution is to not overload any component link, and be able to perform local recovery when one of component link fails.

The network operator may have other objectives such as placing a bidirectional flow or LSP on the same component link in both direction, load balance over component links, composite link energy saving, and etc. These new requirements are described in [I-D.ietf-rtgwg-cl-requirement].

Examples of means to identify a flow may in principle include:

1. an LSP identified by an MPLS label,

2. a sub-LSP [I-D.kompella-mpls-rsvp-ecmp] identified by an MPLS label,

3. a pseudowire (PW) [RFC3985] identified by an MPLS PW label,

4. a flow or group of flows within a pseudowire (PW) [RFC6391] identified by an MPLS flow label,

5. a flow or flow group in an LSP [I-D.ietf-mpls-entropy-label] identified by an MPLS entropy label,

6. all traffic between a pair of IP hosts, identified by an IP source and destination pair,

7. a specific connection between a pair of IP hosts, identified by an IP source and destination pair, protocol, and protocol port pair,

8. a layer-2 conversation within a pseudowire (PW), where the identification is PW payload type specific, such as Ethernet MAC addresses and VLAN tags within an Ethernet PW (RFC4448).

Although in principle a layer-2 conversation within a pseudowire (PW), may be identified by PW payload type specific information, in

practice this is impractical at LSP midpoints when PW are carried. The PW ingress may provide equivalent information in a PW flow label [RFC6391]. Therefore, in practice, item #8 above is covered by [RFC6391] and may be dropped from the list.

An LSR must at least be capable of identifying flows based on MPLS labels. Most MPLS LSP do not require that traffic carried by the LSP are carried in order. MPLS-TP is a recent exception. If it is assumed that no LSP require strict packet ordering of the LSP itself (only of flows within the LSP), then the entire label stack can be used as flow identification. If some LSP may require strict packet ordering but those LSP cannot be distinguished from others, then only the top label can be used as a flow identifier. If only the top label is used (for example, as specified by [RFC4201] when the "all-ones" component described in [RFC4201] is not used), then there may not be adequate flow granularity to accomplish well balanced traffic distribution and it will not be possible to carry LSP that are larger than any individual component link.

The number of flows can be extremely large. This may be the case when the entire label stack is used and is always the case when IP addresses are used in provider networks carrying Internet traffic. Current practice for native IP load balancing at the time of writing were documented in [RFC2991], [RFC2992]. These practices as described, make use of IP addresses. The common practices were extended to include the MPLS label stack and the common practice of looking at IP addresses within the MPLS payload. These extended practices are described in [RFC4385] and [RFC4928] due to their impact on pseudowires without a PWE3 Control Word. Additional detail on current multipath practices can be found in the appendices of [I-D.symmvo-rtgwg-cl-use-cases].

Using only the top label supports too coarse a traffic balance. Using the full label stack or IP addresses as flow identification provides a sufficiently fine traffic balance, but is capable of identifying such a high number of distinct flows, that a technique of grouping flows, such as hashing on the flow identification criteria, becomes essential to reduce the stored state, and is an essential scaling technique. Other means of grouping flows may be possible.

In summary:

1. Load balancing using only the MPLS label stack provides too coarse a granularity of load balance.

2. Tracking every flow is not scalable due to the extremely large number of flows in provider networks.

3. Existing techniques, IP source and destination hash in particular, have proven in over two decades of experience to be an excellent way of identifying groups of flows.

4. If a better way to identify groups of flows is discovered, then that method can be used.

5. IP address hashing is not required, but use of this technique is strongly encouraged given the technique's long history of successful deployment.

2.2. Composite Link in Control Plane

A composite Link is advertised as a single logical interface between two connected routers, which forms forwarding adjacency (FA) between the routers. The FA is advertised as a TE-link in a link state IGP, using either OSPF-TE or ISIS-TE. The IGP-TE advertised interface parameters for the composite link can be preconfigured by the network operator or be derived from its component links. Composite link advertisement requirements are specified in [I-D.ietf-rtgwg-cl-requirement].

In IGP-TE, a composite link is advertised as a single TE link between two connected routers. This is similar to a link bundle [RFC4201]. Link bundle applies to a set of homogenous component links. Composite link allows homogenous and non-homogenous component links. Due to the similarity, and for backwards compatibility, extending link bundling is viewed as both simple and as the best approach.

In order for a route computation engine to calculate a proper path for a LSP, it is necessary for composite link to advertise the summarized available bandwidth as well as the maximum bandwidth that can be made available for single flow (or single LSP where no finer flow identification is available). If a composite link contains some non-homogeneous component links, the composite link also should advertise the summarized bandwidth and the maximum bandwidth for single flow per each homogeneous component link group.

Both LDP [RFC5036] and RSVP-TE [RFC3209] can be used to signal a LSP over a composite link. LDP cannot be extended to support traffic engineering capabilities [RFC3468].

When an LSP is signaled using RSVP-TE, the LSP MUST be placed on the component link that meets the LSP criteria indicated in the signaling message.

When an LSP is signaled using LDP, the LSP MUST be placed on the component link that meets the LSP criteria, if such a component link

is available. LDP does not support traffic engineering capabilities, imposing restrictions on LDP use of Composite Link. See Section 4.2.5 for further details.

A composite link may contain non-homogeneous component links. The route computing engine may select one group of component links for a LSP. The routing protocol MUST make this grouping available in the TE-LSDB. The route computation used in RSVP-TE MUST be extended to include only the capacity of groups within a composite link which meet LSP criteria. The signaling protocol MUST be able to indicate either the criteria, or which groups may be used. A composite link MUST place the LSP on a component link or group which meets or exceeds the LSP criteria.

Composite link capacity is aggregated capacity. LSP capacity MAY be larger than individual component link capacity. Any aggregated LSP can determine a bounds on the largest microflow that could be carried and this constraint can be handled as follows.

1. If no information is available through signaling, management plane, or configuration, the largest microflow is bound by one of the following:

   A. the largest single LSP if most traffic is RSVP-TE signaled and further aggregated,

   B. the largest pseudowire if most traffic is carrying pseudowire payloads that are aggregated within RSVP-TE LSP,

   C. or the largest source and sink interface if a large amount of IP or LDP traffic is contained within the aggregate.

   If a very large amount of traffic being aggregated is IP or LDP, then the largest microflow is bound by the largest component link on which IP traffic can arrive. For example, if an LSR is acting as an LER and IP and LDP traffic is arrving on 10 Gb/s edge interfaces, then no microflow larger than 10 Gb/s will be present on the RSVP-TE LSP that aggregate traffic across the core, even if the core interfaces are 100 Gb/s interfaces.

2. The prior conditions provide a bound on the largest microflow when no signaling extensions indicate a bounds. If an LSP is aggregating smaller LSP for which the largest expected microflow carried by the smaller LSP is signaled, then the largest microflow expected in the containing LSP (the aggregate) is the maximum of the largest expected microflow for any contained LSP. For example, RSVP-TE LSP may be large but aggregate traffic for which the source or sink are all 1 Gb/s or smaller interfaces

(such as in mobile applications in which cell sites backhauls are no larger than 1 Gb/s). If this information is carried in the LSP originated at the cell sites, then further aggregates across a core may make use of this information.

3. The IGP must provide the bounds on the largest microflow that a composite link can accommodate, which is the maximum capacity on a component link that can be made available by moving other traffic. This information is needed by the ingress LER for path determination.

4. A means to signal an LSP whose capacity is larger than individual component link capacity is needed [I-D.ietf-rtgwg-cl-requirement] and also signal the largest microflow expected to be contained in the LSP. If a bounds on the largest microflow is not signaled there is no means to determine if an LSP which is larger than any component link can be subdivided into flows and therefore should be accepted by admission control.

When a bidirectional LSP request is signaled over a composite link, if the request indicates that the LSP must be placed on the same component link, the routers of the composite link MUST place the LSP traffic in both directions on a same component link. This is particularly challenging for aggregated capacity which makes use of the label stack for traffic distribution. The two requirements are mutually exclusive for any one LSP. No one LSP may be both larger than any individual component link and require symmetrical paths for every flow. Both requirements can be accommodated by the same composite link for different LSP, with any one LSP requiring no more than one of these two features.

Individual component link may fail independently. Upon component link failure, a composite link MUST support a minimally disruptive local repair, preempting any LSP which can no longer be supported. Available capacity in other component links MUST be used to carry impacted traffic. The available bandwidth after failure MUST be advertised immediately to avoid looped crankback.

When a composite link is not able to transport all flows, it preempts some flows based upon local management configuration and informs the control plane on these preempted flows. The composite link MUST support soft preemption [RFC5712]. This action ensures the remaining traffic is transported properly. FR#10 requires that the traffic be restored. FR#12 requires that any change be minimally disruptive. These two requirements are interpreted to include preemption among the types of changes that must be minimally disruptive.

2.3. Composite Link in Data Plane

The data plane must first identify groups of flows. Flow identification is covered in Section 2.1. Having identified groups of flows the groups must be placed on individual component links. This second step is called traffic distribution or traffic placement. The two steps together are known as traffic balancing or load balancing.

Traffic distribution may be determined by or constrained by control plane or management plane. Traffic distribution may be changed due to component link status change, subject to constraints imposed by either the management plane or control plane. The distribution function is local to the routers in which a composite link belongs to and is not specified here.

When performing traffic placement, a composite link does not differentiate multicast traffic vs. unicast traffic.

In order to maintain scalability, existing data plane forwarding retains state associated with the top label only. The use of flow group identification is in a second step in the forwarding process. Data plane forwarding makes use of the top label to select a composite link, or a group of components within a composite link or for the case where an LSP is pinned (see [RFC4201]), a specific component link. For those LSP for which the LSP selects only the composite link or a group of components within a composite link, the load balancing makes use of the flow group identification.

The most common traffic placement techniques uses the a flow group identification as an index into a table. The table provides an indirection. The number of bits of hash is constrained to keep table size small. While this is not the best technique, it is the most common. Better techniques exist but they are outside the scope of this document and some are considered proprietary.

Requirements to limit frequency of load balancing can be adhered to by keeping track of when a flow group was last moved and imposing a minimum period before that flow group can be moved again. This is straightforward for a table approach. For other approaches it may be less straightforward but is acheivable.

3. Architecture Tradeoffs

Scalability and stability are critical considerations in protocol design where protocols may be used in a large network such as today’s service provider networks. Composite Link is applicable to networks

which are large enough to require that traffic be split over multiple paths. Scalability is a major consideration for networks that reach a capacity large enough to require Composite Link.

Some of the requirements of Composite Link could potentially have a negative impact on scalability. For example, Composite Link requires additional information to be carried in situations where component links differ in some significant way.

3.1. Scalability Motivations

In the interest of scalability information is aggregated in situations where information about a large amount of network capacity or a large amount of network demand provides is adequate to meet requirements. Routing information is aggregated to reduce the amount of information exchange related to routing and to simplify route computation (see Section 3.2).

In an MPLS network large routing changes can occur when a single fault occurs. For example, a single fault may impact a very large number of LSP traversing a given link. As new LSP are signaled to avoid the fault, resources are consumed elsewhere, and routing protocol announcements must flood the resource changes. If protection is in place, there is less urgency to converging quickly. If multiple faults occur that are not covered by shared risk groups (SRG), then some protection may fail, adding urgency to converging quickly even where protection was deployed.

Reducing the amount of information allows the exchange of information during a large routing change to be accomplished more quickly and simplifies route computation. Simplifying route computation improves convergence time after very significant network faults which cannot be handled by preprovisioned or precomputed protection mechanisms. Aggregating smaller LSP into larger LSP is a means to reduce path computation load and reduce RSVP-TE signaling (see Section 3.3).

Neglecting scaling issues can result in performance issues, such as slow convergence. Neglecting scaling in some cases can result in networks which perform so poorly as to become unstable.

3.2. Reducing Routing Information and Exchange

Link bundling at the very least provides a means of aggregating control plane information. Even where the all-ones component link supported by link bundling is not used, the amount of control information is reduced by the average number of component links in a bundle.

Fully deaggregating link bundle information would negate this benefit. If there is a need to deaggregate, such as to distinguish between groups of links within specified ranges of delay, then no more deaggregation than is necessary should be done.

For example, in supporting the requirement for heterogeneous component links, it makes little sense to fully deaggregate link bundles when adding support for groups of component links with common attributes within a link bundle can maintain most of the benefit of aggregation while adequately supporting the requirement to support heterogeneous component links.

Routing information exchange is also reduced by making sensible choices regarding the amount of change to link parameters that require link readvertisement. For example, if delay measurements include queuing delay, then a much more coarse granularity of delay measurement would be called for than if the delay does not include queuing and is dominated by geographic delay (speed of light delay).

3.3. Reducing Signaling Load

Aggregating traffic into very large hierarchical LSP in the core very substantially reduces the number of LSP that need to be signaled and the number of path computations any given LSR will be required to perform when a major network fault occurs.

In the extreme, applying MPLS to a very large network without hierarchy could exceed the 20 bit label space. For example, in a network with 4,000 nodes, with 2,000 on either side of a cutset, would have 4,000,000 LSP crossing the cutset. Even in a degree four cutset, an uneven distribution of LSP across the cutset, or the loss of one link would result in a need to exceed the size of the label space. Among provider networks, 4,000 access nodes is not at all large.

In less extreme cases, having each node terminate hundreds of LSP to achieve a full mesh creates a very large computational load. The time complexity of one CSPF computation is order(N log N), where L is proportional to N, and N and L are the number of nodes and number of links, respectively. If each node must perform order(N) computations when a fault occurs, then the computational load increases as order(N^2 log N) as the number of nodes increases. In practice at the time of writing, this imposes a limit of a few hundred nodes in a full mesh of MPLS LSP before the computational load is sufficient to result in unacceptable convergence times.

Two solutions are applied to reduce the amount of RSVP-TE signaling. Both involve subdividing the MPLS domain into a core and a set of

regions.

3.3.1. Reducing Signaling Load using LDP

LDP can be used for edge-to-edge LSP, using RSVP-TE to carry the LDP intra-core traffic and also optionally also using RSVP-TE to carry the LDP intra-region traffic within each region. LDP does not support traffic engineering, but does support multipoint-to-point (MPTP) LSP, which require less signaling than edge-to-edge RSVP-TE point-to-point (PTP) LSP. A drawback of this approach is the inability to use RSVP-TE protection (FRR or GMPLS protection) against failure of the border LSR sitting at a core/region boundary.

3.3.2. Reducing Signaling Load using Hierarchy

When the number of nodes grows too large, the amount of RSVP-TE signaling can be reduced using the MPLS PSC hierarchy [RFC4206]. A core within the hierarchy can divide the topology into M regions of on average N/M nodes. Within a region the computational load is reduced by more than M^2. Within the core, the computational load generally becomes quite small since M is usually a fairly small number (a few tens of regions) and each region is generally attached to the core in typically only two or three places on average.

Using hierarchy improves scaling but has two consequences. First, hierarchy effectively forces the use of platform label space. When a containing LSP is rerouted, the labels assigned to the contained LSP cannot be changed but may arrive on a different interface. Second, hierarchy results in much larger LSP. These LSP today are larger than any single component link and therefore force the use of the all-ones component in link bundles.

3.3.3. Using Both LDP and RSVP-TE Hierarchy

It is also possible to use both LDP and RSVP-TE hierarchy. MPLS networks with a very large number of nodes may benefit from the use of both LDP and RSVP-TE hierarchy. The two techniques are certainly not mutually exclusive.

3.4. Reducing Forwarding State

Both LDP and MPLS hierarchy have the benefit of reducing the amount of forwarding state. Using the example from Section 3.3, and using MPLS hierarchy, the worst case generally occurs at borders with the core.

For example, consider a network with approximately 1,000 nodes divided into 10 regions. At the edges, each node requires 1,000 LSP

to other edge nodes. The edge nodes also require 100 intra-region LSP. Within the core, if the core has only 3 attachments to each region the core LSR have less than 100 intra-core LSP. At the border cutset between the core and a given region, in this example there are 100 edge nodes with inter-region LSP crossing that cutset, destined to 900 other edge nodes. That yields forwarding state for on the order of 90,000 LSP at the border cutset. These same routers need only reroute well under 200 LSP when a multiple fault occurs, as long as only links are affected and a border LSR does not go down.

In the core, the forwarding state is greatly reduced. If inter-region LSP have different characteristics, it makes sense to make use of aggregates with different characteristics. Rather than exchange information about every inter-region LSP within the intra-core LSP it makes more sense to use multiple intra-core LSP between pairs of core nodes, each aggregating sets of inter-region LSP with common characteristics or common requirements.

3.5. Avoiding Route Oscillation

Networks can become unstable when a feedback loop exists such that moving traffic to a link causes a metric such as delay to increase, which then causes traffic to move elsewhere. For example, the original ARPANET routing used a delay based cost metric and proved prone to route oscillations [DBP].

Delay may be used as a constraint in routing for high priority traffic, where the movement of traffic cannot impact the delay. The safest way to measure delay is to make measurements based on traffic which is prioritized such that it is queued ahead of the traffic which will be affected. This is a reasonable measure of delay for high priority traffic for which constraints have been set which allow this type of traffic to consume only a fraction of link capacities with the remaining capacity available to lower priority traffic.

Any measurement of jitter (delay variation) that is used in route decision is likely to cause oscillation. Jitter that is caused by queuing effects and cannot be measured using a very high priority measurement traffic flow.

It may be possible to find links with constrained queuing delay or jitter using a theoretical maximum or a probability based bound on queuing delay or jitter at a given priority based on the types and amounts of traffic accepted and combining that theoretical limit with a measured delay at very high priority.

Instability can occur due to poor performance and interaction with protocol timers. In this way a computational scaling problem can

become a stability problem when a network becomes sufficiently large. For this reason, [I-D.ietf-rtgwg-cl-requirement] has a number of requirements focusing on minimally impacting scalability.

4. New Challenges

New technical challenges are posed by [I-D.ietf-rtgwg-cl-requirement] in both the control plane and data plane.

Among the more difficult challenges are the following.

1. requirements related delay or jitter (see Section 4.1.1),

2. the combination of ingress control over LSP placement and retaining an ability to move traffic as demands dictate can pose challenges and such requirements can even be conflicting (see target="sect.local-control" />),

3. path symmetry requires extensions and is particularly challenging for very large LSP (see Section 4.1.3),

4. accommodating a very wide range of requirements among contained LSP can lead to inefficiency if the most stringent requirements are reflected in aggregates, or reduce scalability if a large number of aggregates are used to provide a too fine a reflection of the requirements in the contained LSP (see Section 4.1.4),

5. backwards compatibility is somewhat limited due to the need to accommodate legacy multipath interfaces which provide too little information regarding their configured default behavior, and legacy LSP which provide too little information regarding their requirements (see Section 4.1.5),

6. data plane challenges include those of accommodating very large LSP, large microflows, traffic ordering constraints imposed by a subsent of LSP, and accounting for IP and LDP traffic (see Section 4.2).

4.1. Control Plane Challenges

Some of the control plane requirements are particularly challenging. Handling large flows which aggregate smaller flows must be accomplished with minimal impact on scalability. Potentially conflicting are requirements for jitter and requirements for stability. Potentially conflicting are the requirements for ingress control of a large number of parameters, and the requirements for local control needed to achieve traffic balance across a composite

link. These challenges and potential solutions are discussed in the following sections.

4.1.1. Delay and Jitter Sensitive Routing

Delay and jitter sensitive routing are called for in [I-D.ietf-rtgwg-cl-requirement] in requirements FR#2, FR#7, FR#8, FR#9, FR#15, FR#16, FR#17, FR#18. Requirement FR#17 is particularly problematic, calling for constraints on jitter.

A tradeoff exists between scaling benefits of aggregating information, and potential benefits of using a finer granularity in delay reporting. To maintain the scaling benefit, measured link delay for any given composite link SHOULD be aggregated into a small number of delay ranges. IGP-TE extensions MUST be provided which advertise the available capacities for each of the selected ranges.

For path selection of delay sensitive LSP, the ingress SHOULD bias link metrics based on available capacity and select a low cost path which meets LSP total path delay criteria. To communicate the requirements of an LSP, the ERO MUST be extended to indicate the per link constraints. To communicate the type of resource used, the RRO SHOULD be extended to carry an identification of the group that is used to carry the LSP at each link bundle hop.

4.1.2. Local Control of Traffic Distribution

Many requirements in [I-D.ietf-rtgwg-cl-requirement] suggest that a node immediately adjacent to a component link should have a high degree of control over how traffic is distributed, as long as network performance objectives are met. Particularly relevant are FR#18 and FR#19.

The requirements to allow local control are potentially in conflict with requirement FR#21 which gives full control of component link select to the LSP ingress. While supporting this capability is mandatory, use of this feature is optional per LSP.

A given network deployment will have to consider this pair of conflicting requirements and make appropriate use of local control of traffic placement and ingress control of traffic placement to best meet network requirements.

4.1.3. Path Symmetry Requirements

Requirement FR#21 in [I-D.ietf-rtgwg-cl-requirement] includes a provision to bind both directions of a bidirectional LSP to the same component. This is easily achieved if the LSP is directly signaled

across a composite link. This is not as easily achieved if a set of LSP with this requirement are signaled over a large hierarchical LSP which is in turn carried over a composite link. The basis for load distribution in such as case is the label stack. The labels in either direction are completely independent.

This could be accommodated if the ingress, egress, and all midpoints of the hierarchical LSP make use of an entropy label in the distribution, and use only that entropy label. A solution for this problem may add complexity with very little benefit. There is little or no true benefit of using symmetrical paths rather than component links of identical characteristics.

Traffic symmetry and large LSP capacity are a second pair of conflicting requirements. Any given LSP can meet one of these two requirements but not both. A given network deployment will have to make appropriate use of each of these features to best meet network requirements.

4.1.4. Requirements for Contained LSP

[I-D.ietf-rtgwg-cl-requirement] calls for new LSP constraints. These constraints include frequency of load balancing rearrangement, delay and jitter, packet ordering constraints, and path symmetry.

When LSP are contained within hierarchical LSP, there is no signaling available at midpoint LSR which identifies the contained LSP let alone providing the set of requirements unique to each contained LSP. Defining extensions to provide this information would severely impact scalability and defeat the purpose of aggregating control information and forwarding information into hierarchical LSP. For the same scalability reasons, not aggregating at all is not a viable option for large networks where scalability and stability problems may occur as a result.

As pointed out in Section 4.1.3, the benefits of supporting symmetric paths among LSP contained within hierarchical LSP may not be sufficient to justify the complexity of supporting this capability.

A scalable solution which accommodates multiple sets of LSP between given pairs of LSR is to provide multiple hierarchical LSP for each given pair of LSR, each hierarchical LSP aggregating LSP with common requirements and a common pair of endpoints. This is a network design technique available to the network operator rather than a protocol extension. This technique can accommodate multiple sets of delay and jitter parameters, multiple sets of frequency of load balancing parameters, multiple sets of packet ordering constraints, etc.

4.1.5. Retaining Backwards Compatibility

Backwards compatibility and support for incremental deployment requires considering the impact of legacy LSR in the role of LSP ingress, and considering the impact of legacy LSR advertising ordinary links, advertising Ethernet LAG as ordinary links, and advertising link bundles.

Legacy LSR in the role of LSP ingress cannot signal requirements which are not supported by their control plane software. The additional capabilities supported by other LSR has no impact on these LSR. These LSR however, being unaware of extensions, may try to make use of scarce resources which support specific requirements such as low delay. To a limited extent it may be possible for a network operator to avoid this issue using existing mechanisms such as link administrative attributes and attribute affinities [RFC3209].

Legacy LSR advertising ordinary links will not advertise attributes needed by some LSP. For example, there is no way to determine the delay or jitter characteristics of such a link. Legacy LSR advertising Ethernet LAG pose additional problems. There is no way to determine that packet ordering constraints would be violated for LSP with strict packet ordering constraints, or that frequency of load balancing rearrangement constraints might be violated.

Legacy LSR advertising link bundles have no way to advertise the configured default behavior of the link bundle. Some link bundles may be configured to place each LSP on a single component link and therefore may not be able to accommodate an LSP which requires bandwidth in excess of the size of a component link. Some link bundles may be configured to spread all LSP over the all-ones component. For LSR using the all-ones component link, there is no documented procedure for correctly setting the "Maximum LSP Bandwidth". There is currently no way to indicate the largest microflow that could be supported by a link bundle using the all-ones component link.

Having received the RRO, it is possible for an ingress to look for the all-ones component to identify such link bundles after having signaled at least one LSP. Whether any LSR collects this information on legacy LSR and makes use of it to set defaults, is an implementation choice.

4.2. Data Plane Challenges

Flow identification is briefly discussed in Section 2.1. Traffic distribution is briefly discussed in Section 2.3. This section discusses issues specific to particular requirements specified in

[I-D.ietf-rtgwg-cl-requirement].

4.2.1. Very Large LSP

Very large LSP may exceed the capacity of any single component of a composite link. In some cases contained LSP may exceed the capacity of any single component. These LSP may the use of the equivalent of the all-ones component of a link bundle, or may use a subset of components which meet the LSP requirements.

Very large LSP can be accommodated as long as they can be subdivided (see Section 4.2.2). A very large LSP cannot have a requirement for symetric paths unless complex protocol extensions are proposed (see Section 2.2 and Section 4.1.3).

4.2.2. Very Large Microflows

Within a very large LSP there may be very large microflows. A very large microflow is a very large flows which cannot be further subdivided. Flows which cannot be subdivided must be no larger that the capacity of any single component.

Current signaling provides no way to specify the largest microflow that a can be supported on a given link bundle in routing advertisements. Extensions which address this are discussed in Section 6.4. Absent extensions of this type, traffic containing microflows that are too large for a given composite link may be present. There is no data plane solution for this problem that would not require reordering traffic at the composite link egress.

Some techniques are susceptible to statistical collisions where an algorithm to distribute traffic is unable to disambiguate traffic among two or more very large microflow where their sum is in excess of the capacity of any single component. Hash based algorithms which use too small a hash space are particularly susceptible and require a change in hash seed in the event that this were to occur. A change in hash seed is highly disruptive, causing traffic reordering among all traffic flows over which the hash function is applied.

4.2.3. Traffic Ordering Constraints

Some LSP have strict traffic ordering constraints. Most notable among these are MPLS-TP LSP. In the absence of aggregation into hierarchical LSP, those LSP with strict traffic ordering constraints can be placed on individual component links if there is a means of identifying which LSP have such a constraint. If LSP with strict traffic ordering constraints are aggregated in hierarchical LSP, the hierarchical LSP capacity may exceed the capacity of any single

component link. In such a case the load balancing for the containing may be constrained to look only at the top label and the first contained label. This and related issues are discussed further in Section 6.4.

4.2.4. Accounting for IP and LDP Traffic

Networks which carry RSVP-TE signaled MPLS traffic generally carry low volumes of native IP traffic, often only carrying control traffic as native IP. There is no architectural guarantee of this, it is just how network operators have made use of the protocols.

[I-D.ietf-rtgwg-cl-requirement] requires that native IP and native LDP be accommodated. In some networks, a subset of services may be carried as native IP or carried as native LDP. Today this may be accommodated by the network operator estimating the contribution of IP and LDP and configuring a lower set of available bandwidth figures on the RSVP-TE advertisements.

The only improvement that Composite Link can offer is that of measuring the IP and LDP traffic levels and automatically reducing the available bandwidth figures on the RSVP-TE advertisements. The measurements would have to be significantly filtered. This is similar to a feature in existing LSR, commonly known as "autobandwidth" with a key difference. In the "autobandwidth" feature, the bandwidth request of an RSVP-TE signaled LSP is adjusted in response to traffic measurements. In this case the IP or LDP traffic measurements are used to reduce the link bandwidth directly, without first encapsulating in an RSVP-TE LSP.

This may be a subtle and perhaps even a meaningless distinction if Composite Link is used to form a Sub-Path Maintenance Element (SPME). A SPME is in practice essentially an unsignaled single hop LSP with PHP enabled [RFC5921]. A Composite Link SPME looks very much like classic multipath, where there is no signaling, only management plane configuration creating the multipath entity (of which Ethernet Link Aggregation is a subset).

4.2.5. IP and LDP Limitations

IP does not offer traffic engineering. LDP cannot be extended to offer traffic engineering [RFC3468]. Therefore there is no traffic engineered fallback to an alternate path for IP and LDP traffic if resources are not adequate for the IP and/or LDP traffic alone on a given link in the primary path. The only option for IP and LDP would be to declare the link down. Declaring a link down due to resource exhaustion would reduce traffic to zero and eliminate the resource exhaustion. This would cause oscillations and is therefore not a

viable solution.

Congestion caused by IP or LDP traffic loads is a pathologic case that can occur if IP and/or LDP are carried natively and there is a high volume of IP or LDP traffic. This situation can be avoided by carrying IP and LDP within RSVP-TE LSP.

It is also not possible to route LDP traffic differently for different FEC. LDP traffic engineering is specifically disallowed by [RFC3468]. It may be possible to support multi-topology IGP extensions to accommodate more than one set of criteria. If so, the additional IGP could be bound to the forwarding criteria, and the LDP FEC bound to a specific IGP instance, inheriting the forwarding criteria. Alternately, one IGP instance can be used and the LDP SPF can make use of the constraints, such as delay and jitter, for a given LDP FEC. [Note: WG needs to discuss this and decide first whether to solve this at all and then if so, how.]

5. Existing Mechanisms

In MPLS the one mechanisms which support explicit signaling of multiple parallel links is Link Bundling [RFC4201]. The set of techniques known as "classis multipath" support no explicit signaling, except in two cases. In Ethernet Link Aggregation the Link Aggregation Control Protocol (LACP) coordinates the addition or removal of members from an Ethernet Link Aggregation Group (LAG). The use of the "all-ones" component of a link bundle indicates use of classis multipath, however the ability to determine if a link bundle makes use of classis multipath is not yet supported.

5.1. Link Bundling

Link bundling supports advertisement of a set of homogenous links as a single route advertisement. Link bundling supports placement of an LSP on any single component link, or supports placement of an LSP on the all-ones component link. Not all link bundling implementations support the all-ones component link. There is no way for an ingress LSR to tell which potential midpoint LSR support this feature and use it by default and which do not. Based on [RFC4201] it is unclear how to advertise a link bundle for which the all-ones component link is available and used by default. Common practice is to violate the specification and set the Maximum LSP Bandwidth to the Available Bandwidth. There is no means to determine the largest microflow that could be supported by a link bundle that is using the all-ones component link.

[RFC6107] extends the procedures for hierarchical LSP but also

extends link bundles. An LSP can be explicitly signaled to indicate that it is an LSP to be used as a component of a link bundle. Prior to that the common practice was to simply not advertise the component link LSP into the IGP, since only the ingress and egress of the link bundle needed to be aware of their existence, which they would be aware of due to the RSVP-TE signaling used in setting up the component LSP.

While link bundling can be the basis for composite links, a significant number of small extension needs to be added.

1. To support link bundles of heterogeneous links, a means of advertising the capacity available within a group of homogeneous needs to be provided.

2. Attributes need to be defined to support the following parameters for the link bundle or for a group of homogeneous links.

   A. delay range

   B. jitter (delay variation) range

   C. group metric

   D. all-ones component capable

   E. capable of dynamically balancing load

   F. largest supportable microflow

   G. abilities to support strict packet ordering requirements within contained LSP

3. For each of the prior extended attributes, the constraint based routing path selection needs to be extended to reflect new constraints based on the extended attributes.

4. For each of the prior extended attributes, LSP admission control needs to be extended to reflect new constraints based on the extended attributes.

5. Dynamic load balance must be provided for flows within a given set of links with common attributes such that NPO are not violated including frequency of load balance adjustment for any given flow.

5.2. Classic Multipath

Classic multipath is defined in [I-D.symmvo-rtgwg-cl-use-cases].

Classic multipath refers to the most common current practice in implementation and deployment of multipath. The most common current practice makes use of a hash on the MPLS label stack and if IPv4 or IPv6 are indicated under the label stack, makes use of the IP source and destination addresses [RFC4385] [RFC4928].

Classic multipath provides a highly scalable means of load balancing. Adaptive multipath has proven value in assuring an even loading on component link and an ability to adapt to change in offerred load that occurs over periods of hundreds of milliseconds or more. Classic multipath scalability is due to the ability to effectively work with an extremely large number of flows (IP host pairs) using relatively little resources (a data structure accessed using a hash result as a key or using ranges of hash results).

Classic multipath meets a small subset of Composite Link requirements. Due to scalability of the approach, classic multipath seems to be an excellent candidate for extension to meet the full set of Composite Link forwarding requirements.

Additional detail can be found in [I-D.symmvo-rtgwg-cl-use-cases].

6. Mechanisms Proposed in Other Documents

A number of documents which at the time of writing are works in progress address parts of the requirements of Composite Link, or assist in making some of the goals achievable.

6.1. Loss and Delay Measurement

Procedures for measuring loss and delay are provided in [RFC6374]. These are OAM based measurements. This work could be the basis of delay measurements and delay variation measurement used for metrics called for in [I-D.ietf-rtgwg-cl-requirement].

Currently there are two additional Internet-Drafts that address delay and delay variation metrics.

draft-wang-ccamp-latency-te-metric
: [I-D.wang-ccamp-latency-te-metric] is designed specifically to meet this requirement. OSPF-TE and ISIS-TE extensions are defined to indicate link delay and delay variance. The RSVP-TE ERO is extended to include service level requirements. A latency

accumulation object is defined to provide a means of verification of the service level requirements. This draft is intended to proceed in the CCAMP WG. It is currently and individual submission. The 03 version of this draft expired in September 2012.

draft-giacalone-ospf-te-express-path
This document proposes to extend OSPF-TE only. Extensions support delay, delay variance, loss, residual bandwidth, and available bandwidth. No extensions to RSVP-TE are proposed. This draft is intended to proceed in the CCAMP WG. It is currently and individual submission. The 02 version will expire in March 2012.

A possible course of action may be to combine these two drafts. The delay variance, loss, residual bandwidth, and available bandwidth extensions are particular prone to network instability. The question as to whether queuing delay and delay variation should be considered, and if so for which diffserv Per-Hop Service Class (PSC) is not addressed.

Note to co-authors: The ccamp-latency-te-metric draft refers to [I-D.ietf-rtgwg-cl-requirement] and is well matched to those requirements, including stability. The ospf-te-express-path draft refers to the "Alto Protocol" (draft-ietf-alto-protocol) and therefore may not be intended for RSVP-TE use. The authors of the two drafts may be able to resolve this. It may be best to drop ospf-te-express-path from this framework document.

6.2. Link Bundle Extensions

A set of link bundling extensions are defined in [I-D.ietf-mpls-explicit-resource-control-bundle]. This document provides extensions to the ERO and RRO to explicitly control the labels and resources within a bundle used by an LSP.

The extensions in this document could be further extended to support indicating a group of component links in the ERO or RRO, where the group is given an interface identification like the bundle itself. The extensions could also be further extended to support specification of the all-ones component link in the ERO or RRO.

[I-D.ietf-mpls-explicit-resource-control-bundle] does not provide a means to advertise the link bundle components. It is not certain how the ingress LSR would determine the set of link bundle component links available for a given link bundle.

[I-D.ospf-cc-stlv] provides a baseline draft for extending link

bundling to advertise components. A new component TVL (C-TLV) is proposed, which must reference a Composite Link Link TLV. [I-D.ospf-cc-stlv] is intended for the OSPF WG and submitted for the "Experimental" track. The 00 version expired in February 2012.

6.3. Fat PW and Entropy Labels

Two documents provide a means to add entropy for the purpose of improving load balance. MPLS encapsulation can bury information that is needed to identify microflows. These two documents allow a pseudowire ingress and LSP ingress respectively to add a label solely for the purpose of providing a finer granularity of microflow groups.

[RFC6391] allows pseudowires which carry a large volume of traffic, where microflows can be identified to be load balanced across multiple members of an Ethernet LAG or an MPLS link bundle. This is accomplished by adding a flow label below the pseudowire label in the MPLS label stack. For this to be effective the link bundle load balance must make use of the label stack up to and including this flow label.

[I-D.ietf-mpls-entropy-label] provides a means for a LER to put an additional label known as an entropy label on the MPLS label stack. As defined, only the LER can add the entropy label.

Core LSR acting as LER for aggregated LSP can add entropy labels based on deep packet inspection and place an entropy label indicator (ELI) and entropy label (EL) just below the label being acted on. This would be helpful in situations where the label stack depth to which load distribution can operate is limited by implementation or is limited for other reasons such as carrying both MPLS-TP and MPLS with entropy labels within the same hierarchical LSP.

6.4. Multipath Extensions

The multipath extensions drafts address one aspect of Composite Link. These drafts deal with the issue of accommodating LSP which have strict packet ordering constraints in a network containing multipath. MPLS-TP has become the one important instance of LSP with strict packet ordering constraints and has driven this work.

[I-D.villamizar-mpls-tp-multipath] outlines requirements and gives a number of options for dealing with the apparent incompatibility of MPLS-TP and multipath. A preferred option is described.

[I-D.villamizar-mpls-tp-multipath-te-extn] provides protocol extensions needed to implement the preferred option described in [I-D.villamizar-mpls-tp-multipath].

Other issues pertaining to multipath are also addressed. Means to advertise the largest microflow supportable are defined. Means to indicate the largest expected microflow within an LSP are defined. Issues related to hierarchy are addressed.

7. Required Protocol Extensions and Mechanisms

Prior sections have reviewed key characteristics, architecture tradeoffs, new challenges, existing mechanisms, and relevant mechanisms proposed in existing new documents.

This section first summarizes and groups requirements. A set of documents coverage groupings are proposed with existing works-in-progress noted where applicable. The set of extensions are then grouped by protocol affected as a convenience to implementors.

7.1. Brief Review of Requirements

The following list provides a categorization of requirements specified in [I-D.ietf-rtgwg-cl-requirement] along with a short phrase indication what topic the requirement covers.

routing information aggregation
: FR#1 (routing summarization), FR#20 (composite link may be a component of another composite link)

restoration speed
: FR#2 (restoration speed meeting NPO), FR#12 (minimally disruptive load rebalance), DR#6 (fast convergence), DR#7 (fast worst case failure convergence)

load distribution, stability, minimal disruption
: FR#3 (automatic load distribution), FR#5 (must not oscillate), FR#11 (dynamic placement of flows), FR#12 (minimally disruptive load rebalance), FR#13 (bounded rearrangement frequency), FR#18 (flow placement must satisfy NPO), FR#19 (flow identification finer than per top level LSP), MR#6 (operator initiated flow rebalance)

backward compatibility and migration
: FR#4 (smooth incremental deployment), FR#6 (management and diagnostics must continue to function), DR#1 (extend existing protocols), DR#2 (extend LDP, no LDP TE)

delay and delay variation
: FR#7 (expose lower layer measured delay), FR#8 (precision of latency reporting), FR#9 (limit latency on per LSP basis), FR#15 (minimum delay path), FR#16 (bounded delay path), FR#17 (bounded jitter path)

admission control, preemption, traffic engineering
: FR#10 (admission control, preemption), FR#14 (packet ordering), FR#21 (ingress specification of path), FR#22 (path symmetry), DR#3 (IP and LDP traffic), MR#3 (management specification of path)

single vs multiple domain
: DR#4 (IGP extensions allowed within single domain), DR#5 (IGP extensions disallowed in multiple domain case)

general network management
: MR#1 (polling, configuration, and notification), MR#2 (activation and de-activation)

path determination, connectivity verification
: MR#4 (path trace), MR#5 (connectivity verification)

The above list is not intended as a substitute for [I-D.ietf-rtgwg-cl-requirement], but rather as a concise grouping and reminder or requirements to serve as a means of more easily determining requirements coverage of a set of protocol documents.

7.2. Required Document Coverage

The primary areas where additional protocol extensions and mechanisms are required include the topics described in the following subsections.

There are candidate documents for a subset of the topics below. This grouping of topics does not require that each topic be addressed by a separate document. In some cases, a document may cover multiple topics, or a specific topic may be addressed as applicable in multiple documents.

7.2.1. Component Link Grouping

An extension to link bundling is needed to specify a group of components with common attributes. This can be a TLV defined within the link bundle that carries the same encapsulations as the link bundle. Two interface indices would be needed for each group.

a. An index is needed that if included in an ERO would indicate the need to place the LSP on any one component within the group.

b. A second index is needed that if included in an ERO would indicate the need to balance flows within the LSP across all components of the group. This is equivalent to the "all-ones" component for the entire bundle.

[I-D.ospf-cc-stlv] can be extended to include multipath treatment capabilities. An ISIS solution is also needed. An extension of RSVP-TE signaling is needed to indicate multipath treatment preferences.

If a component group is allowed to support all of the parameters of a link bundle, then a group TE metric would be accommodated. This can be supported with the component TLV (C-TLV) defined in [I-D.ospf-cc-stlv].

The primary focus of this document, among the sets of requirements listed in Section 7.1 is the "routing information aggregation" set of requirements. The "restoration speed", "backward compatibility and migration", and "general network management" requirements must also be considered.

7.2.2. Delay and Jitter Extensions

A extension is needed in the IGP-TE advertisement to support delay and delay variation for links, link bundles, and forwarding adjacencies. Whatever mechanism is described must take precautions that insure that route oscillations cannot occur. [I-D.wang-ccamp-latency-te-metric] may be a good starting point.

The primary focus of this document, among the sets of requirements listed in Section 7.1 is the "delay and delay variation" set of requirements. The "restoration speed", "backward compatibility and migration", and "general network management" requirements must also be considered.

7.2.3. Path Selection and Admission Control

Path selection and admission control changes must be documented in each document that proposes a protocol extension that advertises a new capability or parameter that must be supported by changes in path selection and admission control.

The primary focus of this document, among the sets of requirements listed in Section 7.1 are the "load distribution, stability, minimal disruption" and "admission control, preemption, traffic engineering"

sets of requirements. The "restoration speed" and "path determination, connectivity verification" requirements must also be considered. The "backward compatibility and migration", and "general network management" requirements must also be considered.

7.2.4. Dynamic Multipath Balance

FR#11 explicitly calls for dynamic load balancing similar to existing adaptive multipath. In implementations where flow identification uses a coarse granularity, the adjustments would have to be equally coarse, in the worst case moving entire LSP. The impact of flow identification granularity and potential adaptive multipath approaches may need to be documented in greater detail than provided here.

The primary focus of this document, among the sets of requirements listed in Section 7.1 are the "restoration speed" and the "load distribution, stability, minimal disruption" sets of requirements. The "path determination, connectivity verification" requirements must also be considered. The "backward compatibility and migration", and "general network management" requirements must also be considered.

7.2.5. Frequency of Load Balance

IGP-TE and RSVP-TE extensions are needed to support frequency of load balancing rearrangement called for in FR#13, and FR#15-FR#17. Constraints are not defined in RSVP-TE, but could be modeled after administrative attribute affinities in RFC3209 and elsewhere.

The primary focus of this document, among the sets of requirements listed in Section 7.1 is the "load distribution, stability, minimal disruption" set of requirements. The "path determination, connectivity verification" must also be considered. The "backward compatibility and migration" and "general network management" requirements must also be considered.

7.2.6. Inter-Layer Communication

Lower layer to upper layer communication called for in FR#7 and FR#20. This is addressed for a subset of parameters related to packet ordering in [I-D.villamizar-mpls-tp-multipath] where layers are MPLS. Remaining parameters, specifically delay and delay variation, need to be addressed. Passing information from a lower non-MPLS layer to an MPLS layer needs to be addressed, though this may largely be generic advice encouraging a coupling of MPLS to lower layer management plane or control plane interfaces. This topic can be addressed in each document proposing a protocol extension, where applicable.

The primary focus of this document, among the sets of requirements listed in Section 7.1 is the "restoration speed" set of requirements. The "backward compatibility and migration" and "general network management" requirements must also be considered.

7.2.7. Packet Ordering Requirements

A document is needed to define extensions supporting various packet ordering requirements, ranging from requirements to preservce microflow ordering only, to requirements to preservce full LSP ordering (as in MPLS-TP). This is covered by [I-D.villamizar-mpls-tp-multipath] and [I-D.villamizar-mpls-tp-multipath-te-extn].

The primary focus of this document, among the sets of requirements listed in Section 7.1 are the "admission control, preemption, traffic engineering" and the "path determination, connectivity verification" sets of requirements. The "backward compatibility and migration" and "general network management" requirements must also be considered.

7.2.8. Minimally Disruption Load Balance

The behavior of hash methods used in classic multipath needs to be described in terms of FR#12 which calls for minimally disruptive load adjustments. For example, reseeding the hash violates FR#12. Using modulo operations is significantly disruptive if a link comes or goes down, as pointed out in [RFC2992]. In addition, backwards compatibility with older hardware needs to be accommodated.

The primary focus of this document, among the sets of requirements listed in Section 7.1 is the "load distribution, stability, minimal disruption" set of requirements.

7.2.9. Path Symmetry

Protocol extensions are needed to support dynamic load balance as called for to meet FR#22 (path symmetry) and to meet FR#11 (dynamic placement of flows). Currently path symmetry can only be supported in link bundling if the path is pinned. When a flow is moved both ingress and egress must make the move as close to simultaneously as possible to satisfy FR#22 and FR#12 (minimally disruptive load rebalance). If a group of flows are identified using a hash, then the hash must be identical on the pair of LSR at the endpoint, using the same hash seed and with one side swapping source and destination. If the label stack is used, then either the entire label stack must be a special case flow identification, since the set of labels in either direction are not correlated, or the two LSR must conspire to use the same flow identifier. For example, using a common entropy

label value, and using only the entropy label in the flow identification would satisfy this requirement.

The primary focus of this document, among the sets of requirements listed in Section 7.1 are the "load distribution, stability, minimal disruption" and the "admission control, preemption, traffic engineering" sets of requirements. The "backward compatibility and migration" and "general network management" requirements must also be considered. Path symetry simplifies support for the "path determination, connectivity verification" set of requirements, but with significant complexity added elsewhere.

7.2.10. Performance, Scalability, and Stability

A separate document providing analysis of performance, scalability, and stability impacts of changes may be needed. The topic of traffic adjustment oscillation must also be covered. If sufficient coverage is provided in each document covering a protocol extension, a separate document would not be needed.

The primary focus of this document, among the sets of requirements listed in Section 7.1 is the "restoration speed" set of requirements. This is not a simple topic and not a topic that is well served by scattering it over multiple documents, therefore it may be best to put this in a separate document and put citations in documents called for in Section 7.2.1, Section 7.2.2, Section 7.2.3, Section 7.2.9, Section 7.2.11, Section 7.2.12, Section 7.2.13, and Section 7.2.14. Citation may also be helpful in Section 7.2.4, and Section 7.2.5.

7.2.11. IP and LDP Traffic

A document is needed to define the use of measurements native IP and native LDP traffic levels to reduce link advertised bandwidth amounts.

The primary focus of this document, among the sets of requirements listed in Section 7.1 are the "load distribution, stability, minimal disruption" and the "admission control, preemption, traffic engineering" set of requirements. The "path determination, connectivity verification" must also be considered. The "backward compatibility and migration" and "general network management" requirements must also be considered.

7.2.12. LDP Extensions

Extending LDP is called for in DR#2. LDP can be extended to couple FEC admission control to local resource availability without providing LDP traffic engineering capability. Other LDP extensions

such as signaling a bound on microflow size and LDP LSP requirements would provide useful information without providing LDP traffic engineering capability.

The primary focus of this document, among the sets of requirements listed in Section 7.1 is the "admission control, preemption, traffic engineering" set of requirements. The "backward compatibility and migration" and "general network management" requirements must also be considered.

7.2.13. Pseudowire Extensions

PW extensions such as signaling a bound on microflow size and PW requirements would provide useful information.

The primary focus of this document, among the sets of requirements listed in Section 7.1 is the "admission control, preemption, traffic engineering" set of requirements. The "backward compatibility and migration" and "general network management" requirements must also be considered.

7.2.14. Multi-Domain Composite Link

DR#5 calls for Composite Link to span multiple network topologies. Component LSP may already span multiple network topologies, though most often in practice these are LDP signaled. Component LSP which are RSVP-TE signaled may also span multiple network topologies using at least three existing methods (per domain [RFC5152], BRPC [RFC5441], PCE [RFC4655]). When such component links are combined in a Composite Link, the Composite Link spans multiple network topologies. It is not clear in which document this needs to be described or whether this description in the framework is sufficient. The authors and/or the WG may need to discuss this. DR#5 mandates that IGP-TE extension cannot be used. This would disallow the use of [RFC5316] or [RFC5392] in conjunction with [RFC5151].

The primary focus of this document, among the sets of requirements listed in Section 7.1 are "single vs multiple domain" and "admission control, preemption, traffic engineering". The "routing information aggregation" and "load distribution, stability, minimal disruption" requirements need attention due to their use of the IGP in single domain Composite Link. Other requirements such as "delay and delay variation", can more easily be accomodated by carrying metrics within BGP. The "path determination, connectivity verification" requirements need attention due to requirements to restrict disclosure of topology information across domains in multi-domain deployments. The "backward compatibility and migration" and "general network management" requirements must also be considered.

7.3. Open Issues Regarding Requirements

Note to co-authors: This section needs to be reduced to an empty section and then removed.

The following topics in the requirements document are not addressed. Since they are explicitly mentioned in the requirements document some mention of how they are supported is needed, even if to say nother needed to be done. If we conclude any particular topic is irrelevant, maybe the topic should be removed from the requirement document. At that point we could add the management requirements that have come up and were missed.

1. L3VPN RFC 4364, RFC 4797,L2VPN RFC 4664, VPWS, VPLS RFC 4761, RFC 4762 and VPMS VPMS Framework (draft-ietf-l2vpn-vpms-frmwk-requirements). It is not clear what additional Composite Link requirements these references imply, if any. If no additional requirements are implied, then these references are considered to be informational only.

2. Migration may not be adequately covered in Section 4.1.5. It might also be necessary to say more here on performance, scalability, and stability as it related to migration. Comments on this from co-authors or the WG?

3. We may need a performance section in this document to specifically address #DR6 (fast convergence), and #DR7 (fast worst case failure convergence), though we do already have scalability discussion. The performance section would have to say "no worse than before, except were there was no alternative to make it very slightly worse" (in a bit more detail than that). It would also have to better define the nature of the performance criteria.

7.4. Framework Requirement Coverage by Protocol

As an aid to implementors, this section summarizes requirement coverage listed in Section 7.2 by protocol or LSR functionality affected.

Some documentation may be purely informational, proposing no changes and proposing usage at most. This includes Section 7.2.3, Section 7.2.8, Section 7.2.10, and Section 7.2.14.

Section 7.2.9 may require a new protocol.

7.4.1. OSPF-TE and ISIS-TE Protocol Extensions

Many of the changes listed in Section 7.2 require IGP-TE changes, though most are small extensions to provide additional information. This set includes Section 7.2.1, Section 7.2.2, Section 7.2.5, Section 7.2.6, and Section 7.2.7. An adjustment to existing advertised parameters is suggested in Section 7.2.11.

7.4.2. PW Protocol Extensions

The only suggestion of pseudowire (PW) extensions is in Section 7.2.13.

7.4.3. LDP Protocol Extensions

Potential LDP extensions are described in Section 7.2.12.

7.4.4. RSVP-TE Protocol Extensions

RSVP-TE protocol extensions are called for in Section 7.2.1, Section 7.2.5, Section 7.2.7, and Section 7.2.9.

7.4.5. RSVP-TE Path Selection Changes

Section 7.2.3 calls for path selection to be addressed in individual documents that require change. These changes would include those proposed in Section 7.2.1, Section 7.2.2, Section 7.2.5, and Section 7.2.7.

7.4.6. RSVP-TE Admission Control and Preemption

When a change is needed to path selection, a corresponding change is needed in admission control. The same set of sections applies: Section 7.2.1, Section 7.2.2, Section 7.2.5, and Section 7.2.7. Some resource changes such as a link delay change might trigger preemption. The rules of preemption remain unchanged, still based on holding priority.

7.4.7. Flow Identification and Traffic Balance

The following describe either the state of the art in flow identification and traffic balance or propose changes: Section 7.2.4, Section 7.2.5, Section 7.2.7, and Section 7.2.8.

8. Security Considerations

The security considerations for MPLS/GMPLS and for MPLS-TP are

documented in [RFC5920] and [I-D.ietf-mpls-tp-security-framework].

The types protocol extensions proposed in this framework document provide additional information about links, forwarding adjacencies, and LSP requirements. The protocol semantics changes described in this framework document propose additional LSP constraints applied at path computation time and at LSP admission at midpoints LSR. The additional information and constraints provide no additional security considerations beyond the security considerations already documented in [RFC5920] and [I-D.ietf-mpls-tp-security-framework].

9. Acknowledgments

Authors would like to thank Adrian Farrel, Fred Jounay, Yuji Kamite for his extensive comments and suggestions regarding early versions of this document, Ron Bonica, Nabil Bitar, Eric Gray, Lou Berger, and Kireeti Kompella for their reviews of early versions and great suggestions.

Authors would like to thank Iftekhar Hussain for review and suggestions regarding recent versions of this document.

In the interest of full disclosure of affiliation and in the interest of acknowledging sponsorship, past affiliations of authors are noted. Much of the work done by Ning So occurred while Ning was at Verizon. Much of the work done by Curtis Villamizar occurred while at Infinera. Infinera continues to sponsor this work on a consulting basis.

10. References

10.1. Normative References

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

[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.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC4201] Kompella, K., Rekhter, Y., and L. Berger, "Link Bundling in MPLS Traffic Engineering (TE)", RFC 4201, October 2005.

[RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, October 2005.

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

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5712] Meyer, M. and JP. Vasseur, "MPLS Traffic Engineering Soft Preemption", RFC 5712, January 2010.

[RFC6107] Shiomoto, K. and A. Farrel, "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, February 2011.

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, September 2011.

[RFC6391] Bryant, S., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC 6391, November 2011.

10.2. Informative References

[DBP] Bertsekas, D., "Dynamic Behavior of Shortest Path Routing Algorithms for Communication Networks", IEEE Trans. Auto. Control 1982.

[I-D.ietf-mpls-entropy-label] Drake, J., Kompella, K., Yong, L., Amante, S., and W. Henderickx, "The Use of Entropy Labels in MPLS Forwarding", draft-ietf-mpls-entropy-label-01 (work in progress), October 2011.

[I-D.ietf-mpls-explicit-resource-control-bundle] Zamfir, A., Ali, Z., and P. Dimitri, "Component Link Recording and Resource Control for TE Links", draft-ietf-mpls-explicit-resource-control-bundle-10 (work in progress), April 2011.

[I-D.ietf-mpls-tp-security-framework] Niven-Jenkins, B., Fang, L., Graveman, R., and S. Mansfield, "MPLS-TP Security Framework", draft-ietf-mpls-tp-security-framework-02 (work in progress), October 2011.

[I-D.ietf-rtgwg-cl-requirement]
Malis, A., Villamizar, C., McDysan, D., Yong, L., and N. So, "Requirements for MPLS Over a Composite Link", draft-ietf-rtgwg-cl-requirement-05 (work in progress), January 2012.

[I-D.kompella-mpls-rsvp-ecmp]
Kompella, K., "Multi-path Label Switched Paths Signaled Using RSVP-TE", draft-kompella-mpls-rsvp-ecmp-01 (work in progress), October 2011.

[I-D.ospf-cc-stlv]
Osborne, E., "Component and Composite Link Membership in OSPF", draft-ospf-cc-stlv-00 (work in progress), August 2011.

[I-D.symmvo-rtgwg-cl-use-cases]
Malis, A., Villamizar, C., McDysan, D., Yong, L., and N. So, "Composite Link USe Cases and Design Considerations", draft-symmvo-rtgwg-cl-use-cases-00 (work in progress), February 2012.

[I-D.villamizar-mpls-tp-multipath]
Villamizar, C., "Use of Multipath with MPLS-TP and MPLS", draft-villamizar-mpls-tp-multipath-01 (work in progress), March 2011.

[I-D.villamizar-mpls-tp-multipath-te-extn]
Villamizar, C., "Multipath Extensions for MPLS Traffic Engineering", draft-villamizar-mpls-tp-multipath-te-extn-00 (work in progress), July 2011.

[I-D.wang-ccamp-latency-te-metric]
Fu, X., Betts, M., Wang, Q., McDysan, D., and A. Malis, "GMPLS extensions to communicate latency as a traffic engineering performance metric", draft-wang-ccamp-latency-te-metric-03 (work in progress), March 2011.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, November 2000.

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path

Algorithm", RFC 2992, November 2000.

[RFC3260] Grossman, D., "New Terminology and Clarifications for Diffserv", RFC 3260, April 2002.

[RFC3468] Andersson, L. and G. Swallow, "The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols", RFC 3468, February 2003.

[RFC3945] Mannie, E., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

[RFC3985] Bryant, S. and P. Pate, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, February 2006.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC4928] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", BCP 128, RFC 4928, June 2007.

[RFC5151] Farrel, A., Ayyangar, A., and JP. Vasseur, "Inter-Domain MPLS and GMPLS Traffic Engineering -- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 5151, February 2008.

[RFC5152] Vasseur, JP., Ayyangar, A., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, February 2008.

[RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, December 2008.

[RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, January 2009.

[RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.

[RFC5920] Fang, L., "Security Framework for MPLS and GMPLS Networks", RFC 5920, July 2010.

[RFC5921] Bocci, M., Bryant, S., Frost, D., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, July 2010.

Authors' Addresses

So Ning
Tata Communications

Email: ning.so@tatacommunications.com

Dave McDysan
Verizon
22001 Loudoun County PKWY
Ashburn, VA 20147

Email: dave.mcdysan@verizon.com

Eric Osborne
Cisco

Email: eosborne@cisco.com

Lucy Yong
Huawei USA
5340 Legacy Dr.
Plano, TX 75025

Phone: +1 469-277-5837
Email: lucy.yong@huawei.com

Curtis Villamizar
Outer Cape Cod Network Consulting

Email: curtis@occnc.com

Network Working Group F. Templin, Ed.
Internet-Draft Boeing Research & Technology
Intended status: Standards Track March 14, 2011
Expires: September 15, 2011

Virtual Enterprise Traversal (VET)
draft-templin-intarea-vet-24.txt

Abstract

Enterprise networks connect hosts and routers over various link types, and often also connect to provider networks and/or the global Internet. Enterprise network nodes require a means to automatically provision addresses/prefixes and support internetworking operation in a wide variety of use cases including Small Office, Home Office (SOHO) networks, Mobile Ad hoc Networks (MANETs), ISP networks, multi-organizational corporate networks and the interdomain core of the global Internet itself. This document specifies a Virtual Enterprise Traversal (VET) abstraction for autoconfiguration and operation of nodes in enterprise networks.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

Enterprise networks [RFC4852] connect hosts and routers over various link types (see [RFC4861], Section 2.2). The term "enterprise network" in this context extends to a wide variety of use cases and deployment scenarios. For example, an "enterprise" can be as small as a Small Office, Home Office (SOHO) network, as complex as a multi-organizational corporation, or as large as the global Internet itself. Internet Service Provider (ISP) networks are another example use case that fits well with the VET enterprise network model. Mobile Ad hoc Networks (MANETs) [RFC2501] can also be considered as a challenging example of an enterprise network, in that their topologies may change dynamically over time and that they may employ little/no active management by a centralized network administrative authority. These specialized characteristics for MANETs require careful consideration, but the same principles apply equally to other enterprise network scenarios.

This document specifies a Virtual Enterprise Traversal (VET) abstraction for autoconfiguration and internetworking operation, where addresses of different scopes may be assigned on various types of interfaces with diverse properties. Both IPv4/ICMPv4 [RFC0791][RFC0792] and IPv6/ICMPv6 [RFC2460][RFC4443] are discussed within this context (other network layer protocols are also considered). The use of standard DHCP [RFC2131] [RFC3315] is assumed unless otherwise specified.

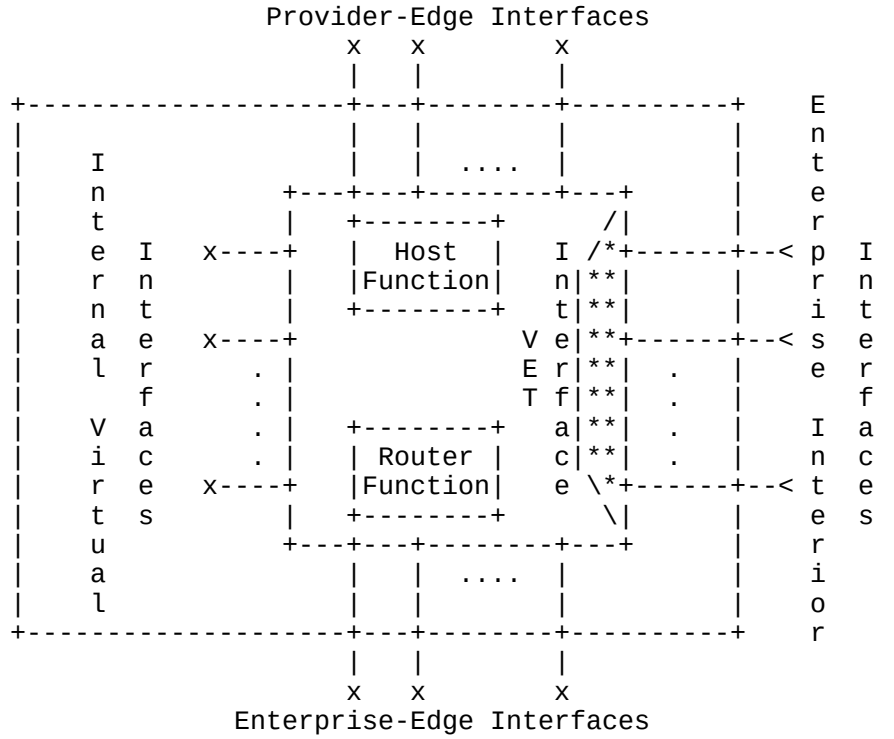


Figure 1: Enterprise Router (ER) Architecture

Figure 1 above depicts the architectural model for an Enterprise Router (ER). As shown in the figure, an ER may have a variety of interface types including enterprise-edge, enterprise-interior, provider-edge, internal-virtual, as well as VET interfaces used for encapsulating inner network layer protocol packets for transmission over outer IPv4 or IPv6 networks. The different types of interfaces are defined, and the autoconfiguration mechanisms used for each type are specified. This architecture applies equally for MANET routers, in which enterprise-interior interfaces typically correspond to the wireless multihop radio interfaces associated with MANETs. Out of scope for this document is the autoconfiguration of provider interfaces, which must be coordinated in a manner specific to the service provider's network.

Enterprise networks require a means for supporting both Provider-(In)dependent (PI) and Provider-Aggregated (PA) addressing. This is especially true for enterprise network scenarios that involve mobility and multihoming. The VET specification provides adaptable mechanisms that address these and other issues in a wide variety of enterprise network use cases.

The VET framework builds on a Non-Broadcast Multiple Access (NBMA) [RFC2491] virtual interface model in a manner similar to other automatic tunneling technologies [RFC2529][RFC5214]. VET interfaces support the encapsulation of inner network layer protocol packets over IP networks (i.e., either IPv4 or IPv6). VET is also compatible with mid-layer encapsulation technologies including IPsec [RFC4301], and supports both stateful and stateless prefix delegation.

VET and its associated technologies (including the Subnetwork Encapsulation and Adaptation Layer (SEAL) [I-D.templin-intarea-seal]) are functional building blocks for a new Internetworking architecture based on the Internet Routing Overlay Network (IRON) [RFC6179] and Routing and Addressing in Networks with Global Enterprise Recursion (RANGER) [RFC5720][RFC6139]. Many of the VET principles can be traced to the deliberations of the ROAD group in January 1992, and also to still earlier initiatives including NIMROD [RFC1753] and the Catenet model for internetworking [CATENET] [IEN48] [RFC2775]. The high-level architectural aspects of the ROAD group deliberations are captured in a "New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG" [RFC1955].

VET is related to the present-day activities of the IETF INTAREA, AUTOCONF, DHC, IPv6, MANET, and V6OPS working groups, as well as the IRTF RRG working group.

2. Terminology

The mechanisms within this document build upon the fundamental principles of IP encapsulation. The term "inner" refers to the innermost {address, protocol, header, packet, etc.} *before* encapsulation, and the term "outer" refers to the outermost {address, protocol, header, packet, etc.} *after* encapsulation. VET also accommodates "mid-layer" encapsulations including the Subnetwork Encapsulation and Adaptation Layer (SEAL) [I-D.templin-intarea-seal], IPsec [RFC4301], etc.

The terminology in the normative references apply; the following terms are defined within the scope of this document:

Virtual Enterprise Traversal (VET)
: an abstraction that uses encapsulation to create virtual overlays for transporting inner network layer packets over outer IPv4 and IPv6 enterprise networks.

enterprise network
: the same as defined in [RFC4852]. An enterprise network is further understood to refer to a cooperative networked collective of devices within a structured IP routing and addressing plan and with a commonality of business, social, political, etc., interests. Minimally, the only commonality of interest in some enterprise network scenarios may be the cooperative provisioning of connectivity itself.

subnetwork
: the same as defined in [RFC3819].

site
: a logical and/or physical grouping of interfaces that connect a topological area less than or equal to an enterprise network in scope. From a network organizational standpoint, a site within an enterprise network can be considered as an enterprise network unto itself.

Mobile Ad hoc Network (MANET)
: a connected topology of mobile or fixed routers that maintain a routing structure among themselves over links that often have dynamic connectivity properties. The characteristics of MANETs are described in [RFC2501], Section 3, and a wide variety of MANETs share common properties with enterprise networks.

enterprise/site/MANET
: throughout the remainder of this document, the term "enterprise network" is used to collectively refer to any of {enterprise, site, MANET}, i.e., the VET mechanisms and operational principles can be applied to enterprises, sites, and MANETs of any size or shape.

VET link
: a virtual link that uses automatic tunneling to create an overlay network that spans an enterprise network routing region. VET links can be segmented (e.g., by filtering gateways) into multiple distinct segments that can be joined together by bridges or IP routers the same as for any link. Bridging would view the multiple (bridged) segments as a single VET link, whereas IP routing would view the multiple segments as multiple distinct VET links. VET links can further be partitioned into multiple logical areas, where each area is identified by a distinct set of border nodes.

VET links configured over non-multicast enterprise networks support only Non-Broadcast, Multiple Access (NBMA) services; VET links configured over enterprise networks that support multicast can support both NBMA and native multicast services. All nodes connected to the same VET link appear as neighbors from the standpoint of the inner network layer.

Enterprise Router (ER)
As depicted in Figure 1, an Enterprise Router (ER) is a fixed or mobile router that comprises a router function, a host function, one or more enterprise-interior interfaces, and zero or more internal virtual, enterprise-edge, provider-edge, and VET interfaces. At a minimum, an ER forwards outer IP packets over one or more sets of enterprise-interior interfaces, where each set connects to a distinct enterprise network.

VET Border Router (VBR)
an ER that connects edge networks to VET links and/or connects multiple VET links together. A VBR is a tunnel endpoint router, and it configures a separate VET interface for each distinct VET link. All VBRs are also ERs.

VET Border Gateway (VBG)
a VBR that connects VET links to provider networks. A VBG may alternately act as "half-gateway", and forward the packets it receives from neighbors on the VET link to another VBG on the same VET link. All VBGs are also VBRs.

VET host
any node (host or router) that configures a VET interface for host-operation only. Note that a node may configure some of its VET interfaces as host interfaces and others as router interfaces.

VET node
any node (host or router) that configures and uses a VET interface.

enterprise-interior interface
an ER’s attachment to a link within an enterprise network. Packets sent over enterprise-interior interfaces may be forwarded over multiple additional enterprise-interior interfaces within the enterprise network before they reach either their final destination or a border router/gateway. Enterprise-interior interfaces connect laterally within the IP network hierarchy.

enterprise-edge interface
a VBR's attachment to a link (e.g., an Ethernet, a wireless personal area network, etc.) on an arbitrarily complex edge network that the VBR connects to a VET link and/or a provider network. Enterprise-edge interfaces connect to lower levels within the IP network hierarchy.

provider-edge interface
a VBR's attachment to the Internet or to a provider network via which the Internet can be reached. Provider-edge interfaces connect to higher levels within the IP network hierarchy.

internal-virtual interface
an interface that is internal to a VET node and does not in itself directly attach to a tangible link, e.g., a loopback interface.

VET interface
a VET node's attachment to a VET link. VET nodes configure each VET interface over a set of underlying enterprise-interior interfaces that connect to a routing region spanned by a single VET link. When there are multiple distinct VET links (each with their own distinct set of underlying interfaces), the VET node configures a separate VET interface for each link.

The VET interface encapsulates each inner packet in any mid-layer headers followed by an outer IP header, then forwards the packet on an underlying interface such that the Time to Live (TTL) - Hop Limit in the inner header is not decremented as the packet traverses the link. The VET interface therefore presents an automatic tunneling abstraction that represents the VET link as a single hop to the inner network layer.

Provider Aggregated (PA) prefix
a network layer protocol prefix that is delegated to a VET node by a provider network.

Provider-(In)dependent (PI) address/prefix
a network layer protocol prefix that is delegated to a VET node by an independent prefix registration authority.

Routing Locator (RLOC)
a public-scope or enterprise-local-scope IP address that can appear in enterprise-interior and/or interdomain routing tables. Public-scope RLOCs are delegated to specific enterprise networks and routable within both the enterprise-interior and interdomain routing regions. Enterprise-local-scope RLOCs (e.g., IPv6 Unique Local Addresses [RFC4193], IPv4 privacy addresses [RFC1918], etc.) are self-generated by individual enterprise networks and routable

only within the enterprise-interior routing region.

ERs use RLOCs for operating the enterprise-interior routing protocol and for next-hop determination in forwarding packets addressed to other RLOCs. End systems can use RLOCs as addresses for end-to-end communications between peers within the same enterprise network. VET interfaces treat RLOCs as *outer* IP addresses during encapsulation.

Endpoint Interface iDentifier (EID)
a public-scope network layer address that is routable within enterprise-edge and/or VET overlay networks. In a pure mapping system, EID prefixes are not routable within the interdomain routing system. In a hybrid routing/mapping system, EID prefixes may be represented within the same interdomain routing instances that distribute RLOC prefixes. In either case, EID prefixes are separate and distinct from any RLOC prefix space, but they are mapped to RLOC addresses to support packet forwarding over VET interfaces.

VBRs participate in any EID-based routing instances and use EID addresses for next-hop determination. End systems can use EIDs as addresses for end-to-end communications between peers either within the same enterprise network or within different enterprise networks. VET interfaces treat EIDs as *inner* network layer addresses during encapsulation.

Note that an EID can also be used as an *outer* network layer address if there are nested encapsulations. In that case, the EID would appear as an RLOC to the innermost encapsulation.

The following additional acronyms are used throughout the document:

CGA - Cryptographically Generated Address
DHCP(v4, v6) - Dynamic Host Configuration Protocol
ECMP - Equal Cost Multi Path
FIB - Forwarding Information Base
ICMP - either ICMPv4 or ICMPv6
IP - either IPv4 or IPv6
ISATAP - Intra-Site Automatic Tunnel Addressing Protocol
NBMA - Non-Broadcast, Multiple Access
ND - Neighbor Discovery
PIO - Prefix Information Option
PRL - Potential Router List
PRLNAME - Identifying name for the PRL
RIB - Routing Information Base
RIO - Route Information Option
SCMP - SEAL Control Message Protocol

SEAL - Subnetwork Encapsulation and Adaptation Layer
SLAAC - IPv6 StateLess Address AutoConfiguration
SNS/SNA - SEAL Neighbor Solicitation/Advertisement
SRS/SRA - SEAL Router Solicitation/Advertisement

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]. When used in lower case (e.g., must, must not, etc.), these words MUST NOT be interpreted as described in [RFC2119], but are rather interpreted as they would be in common English.

3. Enterprise Network Characteristics

Enterprise networks consist of links that are connected by Enterprise Routers (ERs) as depicted in Figure 1. ERs typically participate in a routing protocol over enterprise-interior interfaces to discover routes that may include multiple Layer 2 or Layer 3 forwarding hops. VET Border Routers (VBRs) are ERs that connect edge networks to VET links that span enterprise networks. VET Border Gateways (VBGs) are VBRs that connect VET links to provider networks.

Conceptually, an ER embodies both a host function and router function, and supports communications according to the weak end-system model [RFC1122]. The router function engages in the enterprise-interior routing protocol on its enterprise-interior interfaces, connects any of the ER’s edge networks to its VET links, and may also connect the VET links to provider networks (see Figure 1). The host function typically supports network management applications, but may also support diverse applications typically associated with general-purpose computing platforms.

An enterprise network may be as simple as a small collection of ERs and their attached edge networks; an enterprise network may also contain other enterprise networks and/or be a subnetwork of a larger enterprise network. An enterprise network may further encompass a set of branch offices and/or nomadic hosts connected to a home office over one or several service providers, e.g., through Virtual Private Network (VPN) tunnels. Finally, an enterprise network may contain many internal partitions that are logical or physical groupings of nodes for the purpose of load balancing, organizational separation, etc. In that case, each internal partition resembles an individual segment of a bridged LAN.

Enterprise networks that comprise link types with sufficiently similar properties (e.g., Layer 2 (L2) address formats, maximum transmission units (MTUs), etc.) can configure a subnetwork routing

service such that the inner network layer sees the underlying network as an ordinary shared link the same as for a (bridged) campus LAN (this is often the case with large cellular operator networks). In that case, a single inner network layer hop is sufficient to traverse the underlying network. Enterprise networks that comprise link types with diverse properties and/or configure multiple IP subnets must also provide an enterprise-interior routing service that operates as an IP layer mechanism. In that case, multiple inner network layer hops may be necessary to traverse the underlying network such that care must be taken to avoid multi-link subnet issues [RFC4903].

In addition to other interface types, VET nodes configure VET interfaces that view all other nodes on the VET link as neighbors on a virtual NBMA link. VET nodes configure a separate VET interface for each distinct VET link to which they connect, and discover neighbors on the link that can be used for forwarding packets to off-link destinations. VET interface neighbor relationships may be either unidirectional or bidirectional.

A unidirectional neighbor relationship is typically established and maintained as a result of network layer control protocol messaging in a manner that parallels IPv6 neighbor discovery [RFC4861]. A bidirectional neighbor relationship is typically established and maintained as result of a short transaction between the neighbors carried by a reliable transport protocol such as TCP. The protocol details of the transaction are out of scope for this document, and indeed need not be standardized as long as both neighbors observe the same specifications.

For each distinct VET link , a trust basis must be established and consistently applied. For example, for VET links configured over enterprise networks in which VBRs establish symmetric security associations, mechanisms such as IPsec [RFC4301] can be used to assure authentication and confidentiality. In other enterprise network scenarios, VET links may require asymmetric securing mechanisms such as SEcure Neighbor Discovery (SEND) [RFC3971]. VET links configured over still other enterprise networks may find it sufficient to employ additional encapsulations (e.g., SEAL [I-D.templin-intarea-seal]) that include a simple per-packet nonce to detect off-path attacks.

Finally, for VET links configured over enterprise networks with a centralized management structure (e.g., a corporate campus network, an ISP network, etc.), a hybrid routing/mapping service can be deployed using a synchronized set of VBGs. In that case, the VBGs can provide a "default mapper" [I-D.jen-apt] service used for short-term packet forwarding until route-optimized paths can be established. For VET links configured over enterprise networks with

a distributed management structure (e.g., disconnected MANETs), peer-to-peer coordination between the VET nodes themselves without the assistance of VBGs may be required. Recognizing that various use cases will entail a continuum between a fully centralized and fully distributed approach, the following sections present the mechanisms of Virtual Enterprise Traversal as they apply to a wide variety of scenarios.

4. Autoconfiguration

ERs, VBRs, VBGs, and VET hosts configure themselves for operation as specified in the following subsections.

4.1. Enterprise Router (ER) Autoconfiguration

ERs configure enterprise-interior interfaces and engage in any routing protocols over those interfaces.

When an ER joins an enterprise network, it first configures an IPv6 link-local address on each enterprise-interior interface that requires an IPv6 link-local capability and configures an IPv4 link-local address on each enterprise-interior interface that requires an IPv4 link-local capability. IPv6 link-local address generation mechanisms include Cryptographically Generated Addresses (CGAs) [RFC3972], IPv6 Privacy Addresses [RFC4941], StateLess Address AutoConfiguration (SLAAC) using EUI-64 interface identifiers [RFC4291] [RFC4862], etc. The mechanisms specified in [RFC3927] provide an IPv4 link-local address generation capability.

Next, the ER configures one or more RLOCs and engages in any routing protocols on its enterprise-interior interfaces. The ER can configure RLOCs via administrative configuration, pseudo-random self-generation from a suitably large address pool, DHCP autoconfiguration, or through an alternate autoconfiguration mechanism.

Pseudo-random self-generation of IPv6 RLOCs can be from a large public or local-use IPv6 address range (e.g., IPv6 Unique Local Addresses [RFC4193]). Pseudo-random self-generation of IPv4 RLOCs can be from a large public or local-use IPv4 address range (e.g., [RFC1918]). When self-generation is used alone, the ER continuously monitors the RLOCs for uniqueness, e.g., by monitoring the enterprise-interior routing protocol. (Note however that anycast RLOCs may be assigned to multiple enterprise-interior interfaces; hence, monitoring for uniqueness applies only to RLOCs that are provisioned as unicast.)

DHCP autoconfiguration of RLOCs uses standard DHCP procedures, however ERs acting as DHCP clients SHOULD also use DHCP Authentication [RFC3118] [RFC3315] as discussed further below. In typical enterprise network scenarios (i.e., those with stable links), it may be sufficient to configure one or a few DHCP relays on each link that does not include a DHCP server. In more extreme scenarios (e.g., MANETs that include links with dynamic connectivity properties), DHCP operation may require any ERs that have already configured RLOCs to act as DHCP relays to ensure that client DHCP requests eventually reach a DHCP server. This may result in considerable DHCP message relaying until a server is located, but the DHCP Authentication Replay Detection vector provides relays with a means for avoiding message duplication.

In all enterprise network scenarios, the amount of DHCP relaying required can be significantly reduced if each relay has a way of contacting a DHCP server directly. In particular, if the relay can discover the unicast addresses for one or more servers (e.g., by discovering the unicast RLOC addresses of VBGs as described in Section 4.2.2) it can forward DHCP requests directly to the unicast address(es) of the server(s). If the relay does not know the unicast address of a server, it can forward DHCP requests to a site-scoped DHCP server multicast address if the enterprise network supports site-scoped multicast services. For DHCPv6, relays can forward requests to the site-scoped IPv6 multicast group address 'All_DHCP_Servers' [RFC3315]. For DHCPv4, relays can forward requests to the site-scoped IPv4 multicast group address 'All_DHCPv4_Servers', which SHOULD be set to 239.255.2.1 unless an alternate multicast group for the enterprise network is known. DHCPv4 servers that delegate RLOCs SHOULD therefore join the 'All_DHCPv4_Servers' multicast group and service any DHCPv4 messages received for that group.

A combined approach using both DHCP and self-generation is also possible when the ER configures both a DHCP client and relay that are connected, e.g., via a pair of back-to-back connected Ethernet interfaces, a tun/tap interface, a loopback interface, inter-process communication, etc. The ER first self-generates an RLOC taken from a temporary addressing range used only for the bootstrapping purpose of procuring an actual RLOC taken from a delegated addressing range. The ER then engages in the enterprise-interior routing protocol and performs a DHCP exchange as above using the temporary RLOC as the address of its relay function. When the DHCP server delegates an actual RLOC address/prefix, the ER abandons the temporary RLOC and re-engages in the enterprise-interior routing protocol using an RLOC taken from the delegation.

Alternatively (or in addition to the above), the ER can request RLOC

prefix delegations via an automated prefix delegation exchange over an enterprise-interior interface and can assign the prefix(es) on enterprise-edge interfaces. Note that in some cases, the same enterprise-edge interfaces may assign both RLOC and EID addresses if there is a means for source address selection. In other cases (e.g., for separation of security domains), RLOCs and EIDs are assigned on separate sets of enterprise-edge interfaces.

In some enterprise network scenarios (e.g., MANETs that include links with dynamic connectivity properties), assignment of RLOCs on enterprise-interior interfaces as singleton addresses (i.e., as addresses with /32 prefix lengths for IPv4, or as addresses with /128 prefix lengths for IPv6) MAY be necessary to avoid multi-link subnet issues.

4.2. VET Border Router (VBR) Autoconfiguration

VBRs are ERs that configure and use one or more VET interfaces. In addition to the ER autoconfiguration procedures specified in Section 4.1, VBRs perform the following autoconfiguration operations.

4.2.1. VET Interface Initialization

VBRs configure a separate VET interface for each VET link, where each VET link spans a distinct sets of underlying links belonging to the same enterprise network. All nodes on the VET link appear as single-hop neighbors from the standpoint of the inner network layer protocol through the use of encapsulation.

The VBR binds each VET interface to one or more underlying interfaces, and uses the underlying interface addresses as RLOCs to serve as the outer source addresses for encapsulated packets. The VBR then assigns a link-local address to each VET interface if necessary. When IPv6 and IPv4 are used as the inner/outer protocols (respectively), the VBR can autoconfigure an IPv6 link-local address on the VET interface using a modified EUI-64 interface identifier based on an IPv4 RLOC address (see Section 2.2.1 of [RFC5342]). Link-local address configuration for other inner/outer protocol combinations is through administrative configuration, random self-generation (e.g., [RFC4941], etc.) or through an unspecified alternate method.

4.2.2. Potential Router List (PRL) Discovery

After initializing the VET interface, the VBR next discovers a Potential Router List (PRL) for the VET link that includes the RLOC addresses of VBGs. The PRL can be discovered through administrative configuration, information conveyed in the enterprise-interior

routing protocol, an anycast VBG discovery message exchange, a DHCP option, etc. In multicast-capable enterprise networks, VBRs can also listen for advertisements on the 'rasadv' [RASADV] multicast group address.

When no other information is available, the VBR can resolve an identifying name for the PRL ('PRLNAME') formed as 'hostname.domainname', where 'hostname' is an enterprise-specific name string and 'domainname' is an enterprise-specific Domain Name System (DNS) suffix [RFC1035]. The VBR discovers 'PRLNAME' through administrative configuration, the DHCP Domain Name option [RFC2132], 'rasadv' protocol advertisements, link-layer information (e.g., an IEEE 802.11 Service Set Identifier (SSID)), or through some other means specific to the enterprise network. The VBR can also obtain 'PRLNAME' as part of an arrangement with a private-sector PI prefix vendor (see: Section 4.2.4).

In the absence of other information, the VBR sets the 'hostname' component of 'PRLNAME' to "isatapv2" and sets the 'domainname' component to an enterprise-specific DNS suffix (e.g., "example.com"). Isolated enterprise networks that do not connect to the outside world may have no enterprise-specific DNS suffix, in which case the 'PRLNAME' consists only of the 'hostname' component. (Note that the default hostname "isatapv2" is intentionally distinct from the convention specified in [RFC5214].)

After discovering 'PRLNAME', the VBR resolves the name into a list of RLOC addresses through a name service lookup. For centrally managed enterprise networks, the VBR resolves 'PRLNAME' using an enterprise-local name service (e.g., the DNS). For enterprises with no centralized management structure, the VBR resolves 'PRLNAME' using Link-Local Multicast Name Resolution (LLMNR) [RFC4795] over the VET interface. In that case, all VBGs in the PRL respond to the LLMNR query, and the VBR accepts the union of all responses.

4.2.3. Provider-Aggregated (PA) EID Prefix Autoconfiguration

VBRs that connect their enterprise networks to a provider network obtain Provider-Aggregated (PA) EID prefixes through stateful and/or stateless autoconfiguration mechanisms. The stateful and stateless approaches are discussed in the following subsections.

4.2.3.1. Stateful Prefix Delegation

For IPv4, VBRs acquire IPv4 PA EID prefixes through administrative configuration, an automated IPv4 prefix delegation exchange, etc.

For IPv6, VBRs acquire IPv6 PA EID prefixes through administrative

configuration or through DHCPv6 Prefix Delegation exchanges with an VBG acting as a DHCP relay/server. In particular, the VBR (acting as a requesting router) can use DHCPv6 prefix delegation [RFC3633] over the VET interface to obtain prefixes from the VBG (acting as a delegating router). The VBR obtains prefixes using either a 2-message or 4-message DHCPv6 exchange [RFC3315]. When the VBR acts as a DHCPv6 client, it maps the IPv6 "All_DHCP_Relay_Agents_and_Servers" link- scoped multicast address to the VBG's outer RLOC address.

To perform the 2-message exchange, the VBR's DHCPv6 client function can send a Solicit message with an IA_PD option either directly or via the VBR's own DHCPv6 relay function (see Section 4.1). The VBR's VET interface then forwards the message using VET encapsulation (see: Section 5.4) to a VBG which either services the request or relays it further. The forwarded Solicit message will elicit a Reply message from the server containing prefix delegations. The VBR can also propose a specific prefix to the DHCPv6 server per Section 7 of [RFC3633]. The server will check the proposed prefix for consistency and uniqueness, then return it in the Reply message if it was able to perform the delegation.

After the VBR receives IPv4 and/or IPv6 prefix delegations, it can provision the prefixes on enterprise-edge interfaces as well as on other VET interfaces configured over child enterprise networks for which it acts as an VBG. The VBR can also provision the prefixes on enterprise-interior interfaces to service directly-attached hosts on the enterprise-interior link.

The prefix delegations remain active as long as the VBR continues to renew them via the delegating VBG before lease lifetimes expire. The lease lifetime also keeps the delegation state active even if communications between the VBR and delegating VBG are disrupted for a period of time (e.g., due to an enterprise network partition, power failure, etc.). Note however that if the VBR abandons or otherwise loses continuity with the prefixes, it may be obliged to perform network-wide renumbering if it subsequently receives a new and different set of prefixes.

Stateful prefix delegation for non-IP protocols is out of scope.

4.2.3.2. Stateless Prefix Delegation

When IPv6 and IPv4 are used as the inner and outer protocols, respectively, a stateless IPv6 PA prefix delegation capability is available using the mechanisms specified in [RFC5569][RFC5969]. VBRs can use these mechanisms to statelessly configure IPv6 PA prefixes that embed one of the VBR's IPv4 RLOCs.

Using this stateless prefix delegation, if the IPv4 RLOC changes the IPv6 prefix also changes and the VBR is obliged to renumber any interfaces on which sub-prefixes from the delegated prefix are assigned. This method may therefore be most suitable for enterprise networks in which IPv4 RLOC assignments rarely change, or in enterprise networks in which only services that do not depend on a long-term stable IPv6 prefix (e.g., client-side web browsing) are used.

Stateless prefix delegation for other protocol combinations is out of scope.

4.2.4. Provider-(In)dependent (PI) EID Prefix Autoconfiguration

VBRs can acquire Provider (In)dependent (PI) prefixes to facilitate multihoming, mobility and traffic engineering without requiring site-wide renumbering events. These PI prefixes are made available to VBRs through a prefix delegation authority that may or may not be associated with a specific ISP.

VBRs that connect major enterprise networks (e.g., large corporations, academic campuses, ISP networks, etc.) to a parent enterprise network and/or the global Internet can acquire short PI prefixes (e.g., an IPv6 ::/20, an IPv4 /16, etc.) through a registration authority such as the Internet Assigned Numbers Authority (IANA) or a major regional Internet registry. VBRs that connect small enterprise networks (e.g., SOHO networks, MANETs, etc.) to a parent enterprise network can acquire longer PI prefixes through arrangements with a PI prefix delegation vendor.

After a VBR receives PI prefixes, it can sub-delegate portions of the prefixes on enterprise-edge interfaces, on child VET interfaces for which it is configured as a VBG and on enterprise-interior interfaces to service directly-attached hosts on the enterprise-interior link. The VBR can also sub-delegate portions of its PI prefixes to requesting routers connected to child enterprise networks. These requesting routers consider their sub-delegated portions of the PI prefix as PA, and consider the delegating routers as their points of connection to a provider network.

4.3. VET Border Gateway (VBG) Autoconfiguration

VBGs are VBRs that connect VET links configured over child enterprise networks to provider networks via provider-edge interfaces and/or via VET links configured over parent enterprise networks. A VBG may also act as a "half-gateway", in that it may need to forward the packets it receives from neighbors on the VET link via another VBG connected to the same VET link. This arrangement is seen in the IRON [RFC6179]

client/server/relay architecture, in which a server "half-gateway" is a VBG that forwards packets with off-link destinations via a relay "half-gateway" VBG that connects the VET link to the provider network.

VBGs autoconfigure their provider-edge interfaces in a manner that is specific to the provider connections, and they autoconfigure their VET interfaces that were configured over parent VET links using the VBR autoconfiguration procedures specified in Section 4.2. For each of its VET interfaces connected to child VET links, the VBG initializes the interface the same as for an ordinary VBR (see Section 4.2.1). It then arranges to add one or more of its RLOCs associated with the child VET link to the PRL.

VBGs configure a DHCP relay/server on VET interfaces connected to child VET links that require DHCP services. VBGs may also engage in an unspecified anycast VBG discovery message exchange if they are configured to do so. Finally, VBGs respond to LLMNR queries for 'PRLNAME' on VET interfaces connected to VET links that span child enterprise networks with a distributed management structure.

4.4. VET Host Autoconfiguration

Nodes that cannot be attached via a VBR's enterprise-edge interface (e.g., nomadic laptops that connect to a home office via a Virtual Private Network (VPN)) can instead be configured for operation as a simple host on the VET link. Each VET host performs the same enterprise interior interfaces RLOC configuration procedures as specified for ERs in Section 4.1. The VET host next performs the same VET interface initialization and PRL discovery procedures as specified for VBRs in Section 4.2, except that it configures its VET interfaces as host interfaces (and not router interfaces). Note also that a node may be configured as a host on some VET interfaces and as an VBR/VBG on other VET interfaces.

A VET host may receive non-link-local addresses and/or prefixes to assign to the VET interface via DHCP exchanges and/or through information conveyed in Router Advertisements (RAs). If prefixes are provided, however, there must be assurance that either 1) the VET link will not partition, or 2) that each VET host interface connected to the VET link will configure a unique set of prefixes. VET hosts therefore depend on DHCP and/or RA exchanges to provide only addresses/prefixes that are appropriate for assignment to the VET interface according to these specific cases, and depend on the VBGs within the enterprise keeping track of which addresses/prefixes were assigned to which hosts.

When the VET host solicits a DHCP-assigned EID address/prefix over a

(non-multicast) VET interface, it maps the DHCP relay/server multicast inner destination address to the outer RLOC address of a VBG that it has selected as a default router. The VET host then assigns any resulting DHCP-delegated addresses/prefixes to the VET interface for use as the source address of inner packets. The host will subsequently send all packets destined to EID correspondents via a default router on the VET link, and will discover more-specific routes based on any redirect messages it receives.

5. Internetworking Operation

Following the autoconfiguration procedures specified in Section 4, ERs, VBRs, VBGs, and VET hosts engage in normal internetworking operations as discussed in the following sections.

5.1. Routing Protocol Participation

ERs engage in any RLOC-based routing protocols over enterprise-interior interfaces to exchange routing information for forwarding IP packets with RLOC addresses. VBRs and VBGs can additionally engage in any EID-based routing protocols over VET, enterprise-edge and provider-edge interfaces to exchange routing information for forwarding inner network layer packets with EID addresses. Note that any EID-based routing instances are separate and distinct from any RLOC-based routing instances.

VBR/VBG routing protocol participation on non-multicast VET interfaces uses the NBMA interface model, e.g., in the same manner as for OSPF over NBMA interfaces [RFC5340]. (VBR/VBG routing protocol participation on multicast-capable VET interfaces can alternatively use the standard multicast interface model, but this may result in excessive multicast control message overhead.)

VBRs can use the list of VBGs in the PRL (see: Section 4.2.1) as an initial list of neighbors for EID-based routing protocol participation. VBRs can alternatively use the list of VBGs as potential default routers instead of engaging in an EID-based routing protocol instance. In that case, when the VBR forwards a packet via a default router it may receive a redirect message indicating a different VBR as a better next hop.

5.1.1. PI Prefix Routing Considerations

VBRs that connect large enterprise networks to the global Internet advertise their EID PI prefixes directly into the Internet default-free RIB via the Border Gateway Protocol (BGP) [RFC4271] the same as for a major service provider network. VBRs that connect large

enterprise networks to provider networks can instead advertise their EID PI prefixes into the providers' routing system(s) if the provider networks are configured to accept them.

VBRs that connect small enterprise networks to provider networks obtain one or more PI prefixes and register the prefixes with a serving VBG in the PI prefix vendor's network (e.g., through a vendor-specific short http(s) transaction). The PI prefix vendor network then acts as a virtual "home" enterprise network that connects its customer small enterprise networks to the Internet routing system. The customer small enterprise networks in turn appear as mobile components of the PI prefix vendor's network, i.e., the customer networks are always "away from home".

Further details on routing for PI prefixes is discussed in "The Internet Routing Overlay Network (IRON)" [RFC6179] and "Fib Suppression with Virtual Aggregation" [I-D.ietf-grow-va].

5.2. Default Route Configuration and Selection

Configuration of default routes in the presence of VET interfaces must be carefully coordinated according to the inner and outer network protocols. If the inner and outer protocols are different (e.g., IPv6 within IPv4) then default routes of the inner protocol version can be configured with next-hops corresponding to default routers on a VET interface while default routes of the outer protocol version can be configured with next-hops corresponding to default routers on an underlying interface.

If the inner and outer protocols are the same (e.g., IPv4 within IPv4), care must be taken in setting the default route to avoid ambiguity. For example, if default routes are configured on the VET interface then more-specific routes could be configured on underlying interfaces to avoid looping. In a preferred method, however, multiple default routes can be configured with some having next-hops corresponding to (EID-based) default routers on VET interfaces and others having next-hops corresponding to (RLOC-based) default routers on underlying interfaces. In that case, special next-hop determination rules must be used (see: Section 5.4).

5.3. Address Selection

When permitted by policy and supported by enterprise-interior routing, VET nodes can avoid encapsulation through communications that directly invoke the outer IP protocol using RLOC addresses instead of EID addresses for end-to-end communications. For example, an enterprise network that provides native IPv4 intra-enterprise services can provide continued support for native IPv4 communications

even when encapsulated IPv6 services are available for inter-enterprise communications. In other enterprise network scenarios, the use of EID-based communications (i.e., instead of RLOC-based communications) may be necessary and/or beneficial to support address scaling, transparent Network Address Translator (NAT) traversal, security domain separation, site multihoming, traffic engineering, etc. .

VET nodes can use source address selection rules (e.g., based on name service information) to determine whether to use EID-based or RLOC-based addressing. The remainder of this section discusses internetworking operation for EID-based communications using the VET interface abstraction.

5.4. Next Hop Determination

VET nodes perform normal next-hop determination via longest prefix match, and send packets according to the most-specific matching entry in the FIB. If the FIB entry has multiple next-hop addresses, the VBR selects the next-hop with the best metric value. If multiple next hops have the same metric value, the VET node can use Equal Cost Multi Path (ECMP) to forward different flows via different next-hop addresses, where flows are determined, e.g., by computing a hash of the inner packet's source address, destination address and flow label fields.

If the VET node has multiple default routes of the same inner and outer protocol versions, with some corresponding to EID-based default routers and others corresponding to RLOC-based default routers, it must perform source address based selection of a default route. In particular, if the packet's source address is taken from an EID prefix the VET node selects a default route configured over the VET interface; otherwise, it selects a default route configured over an underlying interface.

As a last resort when there is no matching entry in the FIB (i.e., not even default), VET nodes can discover neighbors within the enterprise network through on-demand name service queries for the EID prefix taken from a packet's destination address (or, by some other inner address to outer address mapping mechanism). For example, for the IPv6 destination address '2001:DB8:1:2::1' and 'PRLNAME' "isatapv2.example.com" the VET node can perform a name service lookup for the domain name:
'0.0.1.0.0.0.8.b.d.0.1.0.0.2.ip6.isatapv2.example.com'.

Name-service lookups in enterprise networks with a centralized management structure use an infrastructure-based service, e.g., an enterprise-local DNS. Name-service lookups in enterprise networks

with a distributed management structure and/or that lack an infrastructure-based name service instead use LLMNR over the VET interface.

When LLMNR is used, the VBR that performs the lookup sends an LLMNR query (with the prefix taken from the IP destination address encoded in dotted-nibble format as shown above) and accepts the union of all replies it receives from neighbors on the VET interface. When a VET node receives an LLMNR query, it responds to the query IFF it aggregates an IP prefix that covers the prefix in the query. If the name-service lookup succeeds, it will return RLOC addresses (e.g., in DNS A records) that correspond to neighbors to which the VET node can forward packets.

5.5. VET Interface Encapsulation/Decapsulation

VET interfaces encapsulate inner network layer packets in any necessary mid-layer headers and trailers (e.g., IPsec [RFC4301], etc.) followed by a SEAL header (if necessary) followed by an outer UDP header (if necessary) followed by an outer IP header. Following all encapsulations, the VET interface submits the encapsulated packet to the outer IP forwarding engine for transmission on an underlying interface. The following sections provide further details on encapsulation:

5.5.1. Inner Network Layer Protocol

The inner network layer protocol sees the VET interface as an ordinary network interface, and views the outer network layer protocol as an ordinary L2 transport. The inner- and outer network layer protocol types are mutually independent and can be used in any combination. Inner network layer protocol types include IPv6 [RFC2460] and IPv4 [RFC0791], but they may also include non-IP protocols such as OSI/CLNP [RFC0994][RFC1070][RFC4548].

5.5.2. Mid-Layer Encapsulation

VET interfaces that use mid-layer encapsulations encapsulate each inner network layer packet in any mid-layer headers and trailers as the first step in a potentially multi-layer encapsulation.

5.5.3. SEAL Encapsulation

Following any mid-layer encapsulations, VET interfaces that use SEAL add a SEAL header as specified in [I-D.templin-intarea-seal]. Inclusion of a SEAL header must be applied uniformly between all neighbors on the VET link. Note that when a VET interface sends a SEAL-encapsulated packet to a neighbor that does not use SEAL

encapsulation, it may receive an ICMP "port unreachable" or "protocol unreachable" depending on whether/not an outer UDP header is included.

SEAL encapsulation is used on VET links that require path MTU mitigations due to encapsulation overhead and/or mechanisms for VET interface neighbor coordination. When SEAL encapsulation is used, the VET interface sets the 'Next Header' value in the SEAL header to the IP protocol number associated with either the mid-layer encapsulation or the IP protocol number of the inner network layer (if no mid-layer encapsulation is used). The VET interface sets the other fields in the SEAL header as specified in [I-D.templin-intarea-seal].

5.5.4. Outer UDP Header Encapsulation

Following any mid-layer and/or SEAL encapsulations, VET interfaces that use UDP encapsulation add an outer UDP header. Inclusion of an outer UDP header must be applied uniformly between all neighbors on the VET link. Note that when a VET interface sends a UDP-encapsulated packet to a neighbor that does not recognize the UDP port number, it may receive an ICMP "port unreachable" message.

VET interfaces use UDP encapsulation on VET links that may traverse NATs and/or legacy networking gear (e.g., Equal Cost MultiPath (ECMP) routers, Link Aggregation Gateways (LAGs), etc.) that only recognize well-known network layer protocols. When UDP encapsulation is used, the VET interface encapsulates the mid-layer packet in an outer UDP header then sets the UDP port numbers as specified for the outermost mid-layer protocol (e.g., IPsec [RFC3947][RFC3948], etc.).

When SEAL [I-D.templin-intarea-seal] is used as the outermost mid-layer protocol, the VET interface maintains per-neighbor local and remote UDP port numbers. For bidirectional neighbors, the interface sets the local UDP port number to the value reserved for SEAL and sets the remote UDP port number to the observed UDP source port number in packets that it receives from the neighbor. In cases in which one of the bidirectional neighbors is behind a NAT, this implies that the one behind the NAT initiates the neighbor relationship. If both neighbors have a way of knowing that there are no NATs in the path, then they may select and set port numbers as described for unidirectional neighbors below.

For unidirectional neighbors, the VET interface sets both the local and remote UDP port numbers to the value reserved for SEAL, and additionally selects a small set of dynamic port number values for use as additional local UDP port numbers. The VET interface then selects one of this set of local port numbers for the UDP source port

for each inner packet it sends, where the port number is determined e.g., by a hash calculated over the inner network layer addresses and inner transport layer port numbers. The VET interface uses a hash function of its own choosing when selecting a dynamic port number value, but it should choose a function that provides uniform distribution between the set of values, and it shoud be consistent in the manner in which the hash is applied.

Finally, for VET links configured over IPv4 enterprise networks, the VET interface sets the UDP checksum field to zero. For VET links configured over IPv6 enterprise networks, considerations for setting the UDP checksum are discussed in [I-D.ietf-6man-udpzero].

5.5.5. Outer IP Header Encapsulation

Following any mid-layer, SEAL and/or UDP encapsulations, the VET interface adds an outer IP header. Outer IP header construction is the same as specified for ordinary IP encapsulation (e.g., [RFC2003], [RFC2473], [RFC4213], etc.) except that the "TTL/Hop Limit", "Type of Service/Traffic Class" and "Congestion Experienced" values in the inner network layer header are copied into the corresponding fields in the outer IP header. The VET interface also sets the IP protocol number to the appropriate value for the first protocol layer within the encapsulation (e.g., UDP, SEAL, IPsec, etc.). When IPv6 is used as the outer IP protocol, the VET interface sets the flow label value in the outer IPv6 header the same as described in [I-D.carpenter-flow-ecmp].

5.5.6. Decapsulation

When a VET interface receives an encapsulated packet, it retains the outer headers and processes the SEAL header as specified in [I-D.templin-intarea-seal].

Next, if the packet will be forwarded from the receiving VET interface into a forwarding VET interface, the VET node copies the "TTL/Hop Limit", "Type of Service/Traffic Class" and "Congestion Experienced" values in the outer IP header received on the receiving VET interface into the corresponding fields in the outer IP header to be sent over the forwarding VET interface (i.e., the values are transferred between outer headers and *not* copied from the inner network layer header). This is true even if the packet is forwarded out the same VET interface that it arrived on, and necessary to support diagnostic functions (e.g., traceroute) and avoid looping.

During decapsulation, when the next-hop is via a non-VET interface, the "Congestion Experienced" value in the outer IP header is copied into the corresponding field in the inner network layer header.

5.6. Mobility and Multihoming Considerations

VBRs that travel between distinct enterprise networks must either abandon their PA prefixes that are relative to the "old" network and obtain PA prefixes relative to the "new" network, or somehow coordinate with a "home" network to retain ownership of the prefixes. In the first instance, the VBR would be required to coordinate a network renumbering event on its attached networks using the new PA prefixes [RFC4192][RFC5887]. In the second instance, an adjunct mobility management mechanism is required.

VBRs can retain their PI prefixes as they travel between distinct network points of attachment as long as they continue to refresh their PI prefix to RLOC address mappings with their serving VBG as described in [RFC6179]. (When the VBR moves far from its serving VBG, it can also select a new VBG in order to maintain optimal routing.) In this way, VBRs can update their PI prefix to RLOC mappings in real time and without requiring an adjunct mobility management mechanism.

The VBGs of a multihomed enterprise network participate in a private inner network layer routing protocol instance (e.g., via an interior BGP instance) to accommodate network partitions/merges as well as intra-enterprise mobility events.

5.7. Neighbor Coordination on VET Interfaces using SEAL

VET interfaces that use SEAL use the SEAL Control Message Protocol (SCMP) as specified in Section 4.5 of [I-D.templin-intarea-seal] to coordinate reachability, routing information, and mappings between the inner and outer network layer protocols. SCMP directly parallels the IPv6 Neighbor Discovery (ND) [RFC4191][RFC4861] and ICMPv6 [RFC4443] protocols, but operates from within the tunnel and supports operation for any combinations of inner and outer network layer protocols.

VET and SEAL are specifically designed for encapsulation of inner network layer payloads over outer IPv4 and IPv6 networks as a link layer. VET interfaces that use SCMP therefore require a new Source/ Target Link-Layer Address Option (S/TLLAO) format that encapsulates IPv4 addresses as shown in Figure 2 and IPv6 addresses as shown in Figure 3:

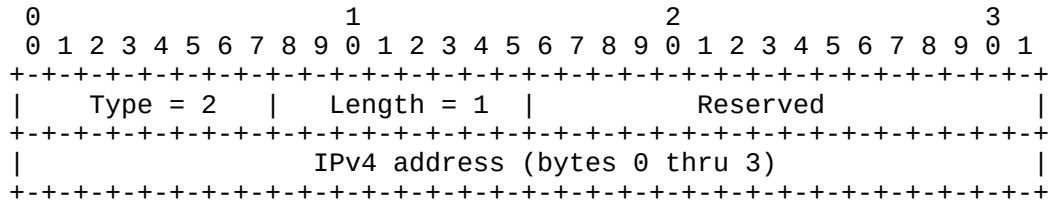


Figure 2: SCMP S/TLLAO Option for IPv4 RLOCs

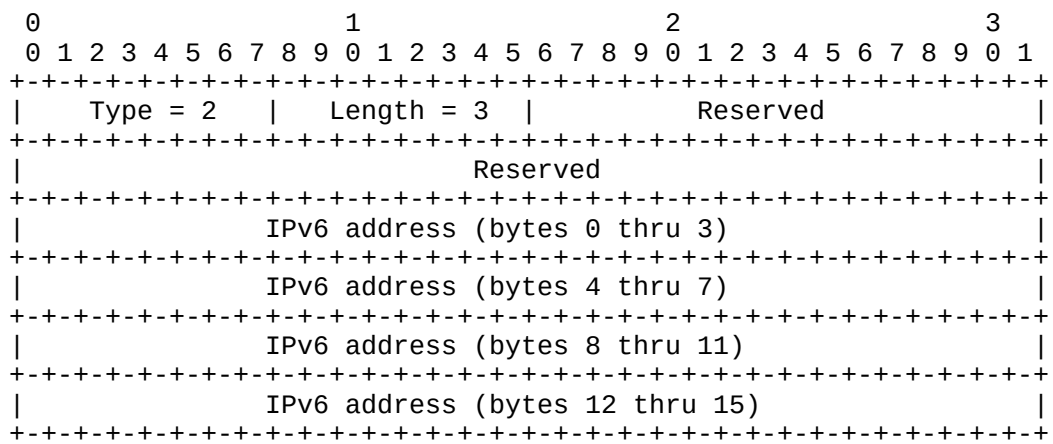


Figure 3: SCMP S/TLLAO Option for IPv6 RLOCs

In addition, VET interfaces that use SCMP use a modified version of the Route Information Option (RIO) (see: [RFC4191]) formatted as shown in Figure 4:

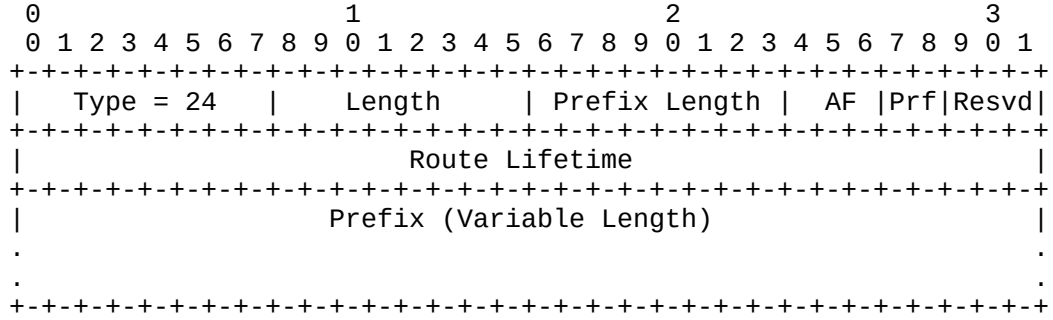


Figure 4: SCMP Route Information Option Format

In this modified format, the VET interface sets the Route Lifetime and Prefix fields in the RIO option the same as specified in

[RFC4191]. It then sets the fields in the header as follows:

- the 'Type', 'Prf', and 'Resvd' fields are set the same as specified in [RFC4191].
- the 'Length' field is set to 1, 2, or 3 as specified in [RFC4191]. It is instead set to 4 if the 'Prefix Length' is greater than 128 and set to 5 if the 'Prefix Length' is greater than 192 (e.g., in order to accommodate longer prefixes of non-IP protocols).
- the 'Prefix Length' field ranges from 0 to 255. The 'Prefix' field is 0, 8, 16, 24 or 32 octets depending on the Length, and the embedded prefix MAY be up to 255 bits in length.
- bits 24 - 26 are used to contain an 'Address Family (AF)' value that indicates the embedded prefix protocol type. This document defines the following values for AF:
  - 000 - IPv4
  - 001 - IPv6
  - 010 - OSI/CLNP NSAP

The following subsections discuss VET interface neighbor coordination using SCMP:

5.7.1. Router Discovery

VET hosts and VBRs can send SCMP Router Solicitation (SRS) messages to one or more VBGs in the PRL to receive solicited SCMP Router Advertisements (SRAs).

When an VBG receives an SRS message on a VET interface, it prepares a solicited SRA message. The SRA includes Router Lifetimes, Default Router Preferences, PIOs and any other options/parameters that the VBG is configured to include. If necessary, the VBG also includes Route Information Options (RIOs) formatted as specified above.

The VBG finally includes one or more SLLAOs formatted as specified above that encode the IPv6 and/or IPv4 RLOC unicast addresses of its own enterprise-interior interfaces or the enterprise-interior interfaces of other nearby VBGs.

5.7.2. Neighbor Unreachability Detection

VET nodes perform Neighbor Unreachability Detection (NUD) on VET interface neighbors by monitoring hints of forward progress enabled

by SEAL mechanisms as evidence that a neighbor is reachable. First, when data packets are flowing, the VET node can periodically set the A bit in the SEAL header of data packets to elicit SCMP responses from the neighbor. Secondly, when no data packets are flowing, the VET node can send periodic probes such as SCMP Neighbor Solicitation (SNS) messages for the same purpose.

Responsiveness to routing changes is directly related to the delay in detecting that a neighbor has gone unreachable. In order to provide responsiveness comparable to dynamic routing protocols, a reasonably short neighbor reachable time (e.g., 5sec) SHOULD be used.

Additionally, a VET node may receive outer IP ICMP "Destination Unreachable; net / host unreachable" messages from an ER on the path indicating that the path to a neighbor may be failing. The node SHOULD first check the packet-in-error to obtain reasonable assurance that the ICMP message is authentic. If the node receives excessive ICMP unreachable errors through multiple RLOCs associated with the same FIB entry, it SHOULD delete the FIB entry and allow subsequent packets to flow through a different route (e.g., a default route with a VBG as the next hop).

5.7.3. Redirect Function

[[UNDER CONSTRUCTION]]

This section will be updated to reflect the new technique known as "Predirection" as discussed for ISATAP updates in Section 5.14.

[[UNDER CONSTRUCTION]]

5.8. Neighbor Coordination on VET Interfaces using IPsec

VET interfaces that use IPsec encapsulation use the Internet Key Exchange protocol, version 2 (IKEv2) [RFC4306] to manage security association setup and maintenance. IKEv2 provides a logical equivalent of the SCMP in terms of VET interface neighbor coordinations; for example, IKEv2 also provides mechanisms for redirection [RFC5685] and mobility [RFC4555].

IPsec additionally provides an extended Identification field and integrity check vector; these features allow IPsec to utilize outer IP fragmentation and reassembly with less risk of exposure to data corruption due to reassembly misassociations. On the other hand, IPsec entails the use of symmetric security associations and hence may not be appropriate to all enterprise network use cases.

5.9. Multicast

5.9.1. Multicast over (Non)Multicast Enterprise Networks

Whether or not the underlying enterprise network supports a native multicasting service, the VET node can act as an inner network layer IGMP/MLD proxy [RFC4605] on behalf of its attached edge networks and convey its multicast group memberships over the VET interface to a VBG acting as a multicast router. Its inner network layer multicast transmissions will therefore be encapsulated in outer headers with the unicast address of the VBG as the destination.

5.9.2. Multicast Over Multicast-Capable Enterprise Networks

In multicast-capable enterprise networks, ERs provide an enterprise-wide multicasting service (e.g., Simplified Multicast Forwarding (SMF) [I-D.ietf-manet-smf], Protocol Independent Multicast (PIM) routing, Distance Vector Multicast Routing Protocol (DVMRP) routing, etc.) over their enterprise-interior interfaces such that outer IP multicast messages of site-scope or greater scope will be propagated across the enterprise network. For such deployments, VET nodes can optionally provide a native inner multicast/broadcast capability over their VET interfaces through mapping of the inner multicast address space to the outer multicast address space. In that case, operation of link-or greater-scoped inner multicasting services (e.g., a link-scoped neighbor discovery protocol) over the VET interface is available, but SHOULD be used sparingly to minimize enterprise-wide flooding.

VET nodes encapsulate inner multicast messages sent over the VET interface in any mid-layer headers (e.g., UDP, SEAL, IPsec, etc.) followed by an outer IP header with a site-scoped outer IP multicast address as the destination. For the case of IPv6 and IPv4 as the inner/outer protocols (respectively), [RFC2529] provides mappings from the IPv6 multicast address space to a site-scoped IPv4 multicast address space (for other encapsulations, mappings are established through administrative configuration or through an unspecified alternate static mapping).

Multicast mapping for inner multicast groups over outer IP multicast groups can be accommodated, e.g., through VET interface snooping of inner multicast group membership and routing protocol control messages. To support inner-to-outer multicast address mapping, the VET interface acts as a virtual outer IP multicast host connected to its underlying interfaces. When the VET interface detects that an inner multicast group joins or leaves, it forwards corresponding outer IP multicast group membership reports on an underlying interface over which the VET interface is configured. If the VET

node is configured as an outer IP multicast router on the underlying interfaces, the VET interface forwards locally looped-back group membership reports to the outer IP multicast routing process. If the VET node is configured as a simple outer IP multicast host, the VET interface instead forwards actual group membership reports (e.g., IGMP messages) directly over an underlying interface.

Since inner multicast groups are mapped to site-scoped outer IP multicast groups, the VET node MUST ensure that the site-scoped outer IP multicast messages received on the underlying interfaces for one VET interface do not "leak out" to the underlying interfaces of another VET interface. This is accommodated through normal site-scoped outer IP multicast group filtering at enterprise network boundaries.

5.10. Service Discovery

VET nodes can perform enterprise-wide service discovery using a suitable name-to-address resolution service. Examples of flooding-based services include the use of LLMNR [RFC4795] over the VET interface or multicast DNS (mDNS) [I-D.cheshire-dnsext-multicastdns] over an underlying interface. More scalable and efficient service discovery mechanisms (e.g., anycast) are for further study.

5.11. VET Link Partitioning

A VET link can be partitioned into multiple distinct logical groupings. In that case, each partition configures its own distinct 'PRLNAME' (e.g., 'isatapv2.zone1.example.com', 'isatapv2.zone2.example.com', etc.).

VBGs can further create multiple IP subnets within a partition, e.g., by sending SRAs with PIOs containing different IP prefixes to different groups of VET hosts. VBGs can identify subnets, e.g., by examining RLOC prefixes, observing the enterprise-interior interfaces over which SRSs are received, etc.

In the limiting case, VBGs can advertise a unique set of IP prefixes to each VET host such that each host belongs to a different subnet (or set of subnets) on the VET interface.

5.12. VBG Prefix State Recovery

VBGs retain explicit state that tracks the inner network layer prefixes delegated to VBRs connected to the VET link, e.g., so that packets are delivered to the correct VBRs. When a VBG loses some or all of its state (e.g., due to a power failure), client VBRs must refresh the VBG's state so that packets can be forwarded over correct

routes.

5.13. Legacy ISATAP Services

VBGs can support legacy ISATAP services according to the specifications in [RFC5214]. In particular, VBGs can configure legacy ISATAP interfaces and VET interfaces over the same sets of underlying interfaces as long as the PRLs and IPv6 prefixes associated with the ISATAP/VET interfaces are distinct.

Legacy ISATAP hosts acquire addresses and/or prefixes in the same manner and using the same mechanisms as described for VET hosts in Section 4.4 above.

In order to support dynamic on-demand routing on ISATAP interfaces, a new (and backwards-compatible) approach called "ISATAP Predirection" is specified in the following sections:

5.14. ISATAP Update

In order to support dynamic on-demand routing on ISATAP interfaces, a new (and backwards-compatible) approach called "ISATAP Predirection" is specified in the following sections. This section updates [RFC5214].

5.14.1. ISATAP Predirection

Figure 5 depicts a reference ISATAP network topology. The scenario shows an advertising ISATAP router (’A’), two non-advertising ISATAP routers (’B’, ’D’) and two ordinary IPv6 hosts (’C’, ’E’) in a typical deployment configuration:

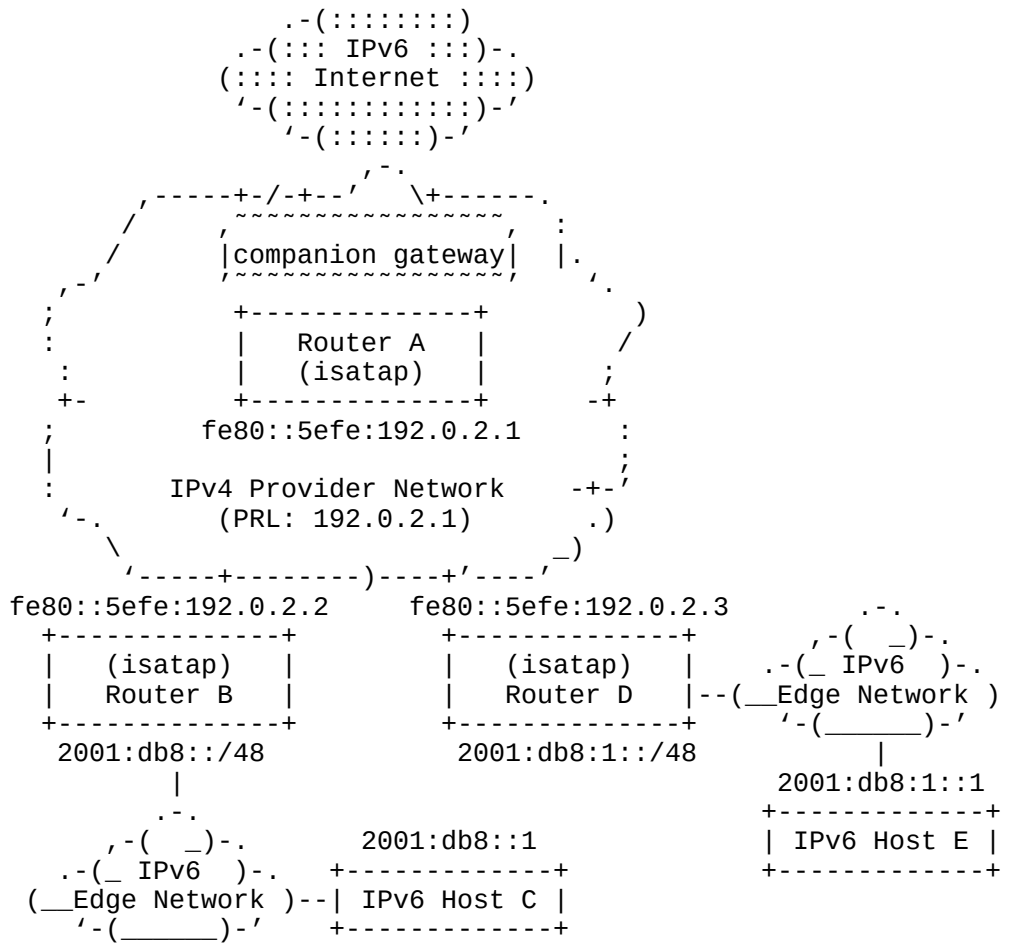


Figure 5: Reference ISATAP Network Topology

With reference to Figure 5, when router 'A' receives an IPv6 packet on an advertising ISATAP interface that it will forward back out the same interface, 'A' must arrange to redirect the originating ISATAP node 'B' to a better next hop ISATAP node 'D' that is closer to the final destination 'E'. First, however, 'A' must direct 'D' to establish a forwarding table entry so that it will have a means to determine that 'B' is authorized to produce packets using a given source address. This process is accommodated via a unidirectional reliable exchange in which 'A' first informs 'D', then 'D' informs 'B' via 'A' as a trusted intermediary. 'B' therefore knows that 'D' will accept the packets it sends as long as 'D' retains the forwarding table entry. We call this process "predirection", which stands in contrast to ordinary IPv6 redirection.

Consider the alternative in which 'A' informs both 'B' and 'D' separately via independent IPv6 Redirect messages (see: [RFC4861]). In that case, several conditions can occur that could result in communications failures. First, if 'B' receives the Redirect message but 'D' does not, subsequent packets sent by 'B' would disappear into a black hole since 'D' would not have a forwarding table entry to verify their source addresses. Second, if 'D' receives the Redirect message but 'B' does not, subsequent packets sent in the reverse direction by 'D' would be lost. Finally, timing issues surrounding the establishment and garbage collection of forwarding table entries at 'B' and 'D' could yield unpredictable behavior. For example, unless the timing were carefully coordinated through some form of synchronization loop, there would invariably be instances in which one node has the correct forwarding table state and the other node does not resulting in non-deterministic packet loss.

The following subsections discuss the predirection steps that support the reference operational scenario:

5.14.1.1. 'A' Sends Predirect Forward To 'D'

When 'A' forwards an original IPv6 packet sent by 'B' out the same ISATAP interface that it arrived on, it sends a "Predirect" message forward toward 'D' instead of sending a Redirect message back to 'B'. The Predirect message is simply an ISATAP-specific version of an ordinary IPv6 Redirect message as depicted in Section 4.5 of [RFC4861], and is identified by two new backward-compatible bits taken from the Reserved field as shown in Figure 6:

Figure 6: ISATAP-Specific IPv6 Redirect Message Format

Where the new bits are defined as:

I (1) the "ISATAP" bit. Set to 1 to indicate an ISATAP-specific Redirect message, and set to 0 to indicate an ordinary IPv6 Redirect message.

P (1) the "Predirect" bit. Set to 1 to indicate a Predirect message, and set to 0 to indicate a Redirect response to a Predirect message. (This bit is valid only when the I bit is set to 1.)

Using this new Predirect message format, 'A' prepares the message in a similar fashion as for an ordinary ISATAP-encapsulated IPv6 Redirect message as follows:

o the outer IPv4 source address is set to 'A's IPv4 address.

o the outer IPv4 destination address is set to 'D's IPv4 address.

- o the inner IPv6 source address is set to 'A's ISATAP link-local address.
- o the inner IPv6 destination address is set to 'D's ISATAP link-local address.
- o the Predirect Target and Destination Addresses are both set to 'B's ISATAP link-local address.
- o the Predirect message includes Route Information Options (RIOs) [RFC4191] that encode an IPv6 prefix taken from 'B's address/ prefix delegations that covers the IPv6 source address of the originating IPv6 packet.
- o the Predirect message includes a Redirected Header Option (RHO) that contains at least the header of the originating IPv6 packet.
- o the I and P bits in the Predirect message header are both set to 1.

'A' then sends the Predirect message forward to 'D'.

5.14.1.2. 'D' Processes the Predirect and Sends Redirect Back To 'A'

When 'D' receives the Predirect message, it decapsulates the message according to Section 7.3 of [RFC5214] since the outer IPv4 source address is a member of the PRL.

'D' then uses the message validation checks specified in Section 8.1 of [RFC4861], except that instead of verifying that the "IP source address of the Redirect is the same as the current first-hop router for the specified ICMP Destination Address" (i.e., the 6th verification check), it accepts the message if the "outer IP source address of the Predirect is the same as the current first-hop router for the destination address of the originating IPv6 packet encapsulated in the RHO". (Note that this represents an ISATAP-specific adaptation of the verification checks.) Finally, 'D' only accepts the message if the destination address of the originating IPv6 packet encapsulated in the RHO is covered by one of its CURRENT delegated addresses/prefixes (see Section 5.14.4).

'D' then either creates or updates an IPv6 forwarding table entry with the prefix encoded in the RIO option as the target prefix, and the IPv6 Target Address of the Predirect message (i.e., 'B's ISATAP link-local address) as the next hop. 'D' places the entry in the FILTERING state, then sets/resets a filtering expiration timer value of 40 seconds. If the filtering timer expires, the node clears the FILTERING state and deletes the forwarding table entry if it is not

in the FORWARDING state. This suggests that 'D's ISATAP interface should maintain a private forwarding table separate from the common IPv6 forwarding table, since the entry must be managed by the ISATAP interface itself.

After processing the Predirect message and establishing the forwarding table entry, 'D' prepares an ISATAP Redirect message in response to the Predirect as follows:

- the outer IPv4 source address is set to 'D's IPv4 address.
- the outer IPv4 destination address is set to 'A's IPv4 address.
- the inner IPv6 source address, is set to 'D's ISATAP link-local address.
- the inner IPv6 destination address is set to 'A's ISATAP link-local address.
- the Redirect Target and the Redirect Destination Addresses are both set to 'D's ISATAP link-local address.
- the Redirect message includes RIOs that encode IPv6 prefixes taken from 'D's address/prefix delegations that covers the IPv6 destination address of the originating IPv6 packet encapsulated in the Redirected Header option of the Predirect.
- the Redirect message includes an RHO copied from the corresponding Predirect message.
- the (I, P) bits in the Redirect message header are set to (1, 0).

'D' then sends the Redirect message to 'A'.

5.14.1.3. 'A' Processes the Redirect then Proxies it Back To 'B'

When 'A' receives the Redirect message, it decapsulates the message according to Section 7.3 of [RFC5214] since the inner IPv6 source address embeds the outer IPv4 source address.

'A' next accepts the message only if it satisfies the same message validation checks specified for Predirects in Section 3.2.4.6.2.

'A' then locates a forwarding table entry that covers the IPv6 source address of the packet segment in the RHO (i.e., a forwarding table entry with next hop 'B'), then proxies the Redirect message back toward 'B'. Without decrementing the IPv6 hop limit in the Redirect message, 'A' next changes the IPv4 source address of the Redirect

message to its own IPv4 address, changes the IPv4 destination address to 'B's IPv4 address, changes the IPv6 source address to its own IPv6 link-local address, and changes the IPv6 destination address to 'B's IPv6 link-local address. 'A' then sends the proxied Redirect message to 'B'.

5.14.1.4. 'B' Processes The Redirect Message

When 'B' receives the Redirect message, it decapsulates the message according to Section 7.3 of [RFC5214] since the outer IPv4 source address is a member of the PRL.

'B' next accepts the message only if it satisfies the same message validation checks specified for Predirects in Section 3.2.4.6.2.

'B' then either creates or updates an IPv6 forwarding table entry with the prefix encoded in the RIO option as the target prefix, and the IPv6 Target Address of the Redirect message (i.e., 'D's ISATAP link-local address) as the next hop. 'B' places the entry in the FORWARDING state, then sets/resets a forwarding expiration timer value of 30 seconds. If the forwarding timer expires, the node clears the FORWARDING state and deletes the forwarding table entry if it is not in the FILTERING state. Again, this suggests that 'B's ISATAP interface should maintain a private forwarding table separate from the common IPv6 forwarding table, since the entry must be managed by the ISATAP interface itself.

Now, 'B' has a forwarding table entry in the FORWARDING state, and 'D' has a forwarding table entry in the FILTERING state. Therefore, 'B' may send ordinary IPv6 data packets with destination addresses covered by 'D's prefix directly to 'D' without involving 'A'. 'D' will in turn accept the packets since it has a forwarding table entry authorizing 'B' to source packets from its claimed IPv6 address.

To enable packet forwarding from 'D' directly to 'B', a reverse-predirection operation is required which is the mirror-image of the forward-predirection operation described above. Following the reverse predirection, both 'B' and 'D' will have forwarding table entries in the "(FORWARDING | FILTERING)" state, and IPv6 packets can be exchanged bidirectionally without involving 'A'.

5.14.1.5. 'B' Sends Periodic Predirect Messages Forward to 'A'

In order to keep forwarding table entries alive while data packets are actively flowing, 'B' can periodically send additional Predirect messages via 'A' to solicit Redirect messages from 'D'. When 'B' forwards an IPv6 packet via 'D', and the corresponding forwarding table entry FORWARDING state timer is nearing expiration, 'B' sends

Predirect messages (subject to rate limiting) prepared as follows:

- the outer IPv4 source address is set to 'B's IPv4 address.
- the outer IPv4 destination address is set to 'A's IPv4 address.
- the inner IPv6 source address is set to 'B's ISATAP link-local address.
- the inner IPv6 destination address is set to 'A's ISATAP link-local address.
- the Predirect Target and Destination Addresses are both set to 'B's ISATAP link-local address.
- the Predirect message includes RIOs that encode IPv6 prefixes taken from 'B's address/prefix delegations that cover the IPv6 source address of the originating IPv6 packet.
- the Predirect message includes an RHO that contains at least the header of the originating IPv6 packet.
- the I and P bits in the Predirect message header are both set to 1.

When 'A' receives the Predirect message, it decapsulates the message according to Section 7.3 of [RFC5214] since the inner IPv6 source address embeds the outer IPv4 source address.

'A' next accepts the message only if it satisfies the same message validation checks specified for Predirects in Section 3.2.4.6.2.

'A' then locates a forwarding table entry that covers the IPv6 destination address of the packet segment in the RHO (in this case, a forwarding table entry with next hop 'D'). Without decrementing the IPv6 hop limit in the Redirect message, 'A' next changes the IPv4 source address of the Predirect message to its own IPv4 address, changes the IPv4 destination address to 'D's IPv4 address, changes the IPv6 source address to its own IPv6 link-local address, and changes the IPv6 destination address to 'D's IPv6 link-local address. 'A' then sends the proxied Predirect message to 'D'. When 'D' receives the proxied message, it processes the message the same as if it had originated from 'A' as described in Section 3.2.4.6.2.

5.14.2. Scaling Considerations

Figure 5 depicts an ISATAP network topology with only a single advertising ISATAP router within the provider network. In order to

support larger numbers of non-advertising ISATAP routers and ISATAP hosts, the provider network can deploy more advertising ISATAP routers to support load balancing and generally shortest-path routing.

Such an arrangement requires that the advertising ISATAP routers participate in an IPv6 routing protocol instance so that IPv6 address/prefix delegations can be mapped to the correct router. The routing protocol instance can be configured as either a full mesh topology involving all advertising ISATAP routers, or as a partial mesh topology with each ISATAP router associating with one or more companion gateways and a full mesh between companion gateways.

5.14.3. Proxy Chaining

In large ISATAP deployments, there may be many advertising ISATAP routers, each serving many ISATAP clients (i.e., both non-advertising routers and simple hosts). The advertising ISATAP routers then either require full topology knowledge, or a default route to a companion gateway that does have full topology knowledge. For example, if Client 'A' connects to advertising ISATAP router 'B', and Client 'E' connects to advertising ISATAP router 'D', then 'B' and 'D' must either have full topology knowledge or have a default route to a companion gateway (e.g., 'C') that does.

In that case, when 'A' sends an initial packet to 'E', 'B' generates a Predirect message toward 'C', which proxies the message toward 'D' which finally proxies the message toward 'E'.

In the reverse direction, when 'E' sends a Redirect response message to 'A', it first sends the message to 'D', which proxies the message toward 'C', which proxies the message toward 'B', which finally proxies the message toward 'A'.

5.14.4. Mobility

An ISATAP router 'A' can configure both a non-advertising ISATAP interface on a provider network and an advertising ISATAP interface on an edge network. In that case, 'A' can service ISATAP clients (i.e. both non-advertising routers and simple hosts) within the edge network by acting as a DHCPv6 relay. When a client 'B' in the edge network that has obtained IPv6 addresses/prefixes moves to a different edge network, however, 'B' can release its address/prefix delegations via 'A' and re-establish them via a different ISATAP router 'C' in the new edge network.

When 'B' releases its address/prefix delegations via 'A', 'A' marks the IPv6 forwarding table entries that cover the addresses/prefixes

as DEPARTED (i.e., it clears the CURRENT state). 'A' therefore ceases to respond to Predirect messages correlated with the DEPARTED entries, and also schedules a garbage-collection timer of 60 seconds, after which it deletes the DEPARTED entries.

When 'A' receives IPv6 packets destined to an address covered by the DEPARTED IPv6 forwarding table entries, it forwards them to the last-known edge network link-layer address of 'B' as a means for avoiding mobility-related packet loss during routing changes. Eventually, correspondents will receive new Redirect messages from the network to discover that 'B' is now associated with 'C'.

Note that this mobility management method works the same way when the edge networks comprise native IPv6 links (i.e., and not just for ISATAP links), however any IPv6 packets forwarded by 'A' via an IPv6 forwarding table entry in the DEPARTED state may be lost if the mobile node moves off-link with respect to its previous edge network point of attachment. This should not be a problem for large links (e.g., large cellular network deployments, large ISP networks, etc.) in which all/most mobility events are intra-link.

6. IANA Considerations

There are no IANA considerations for this document.

7. Security Considerations

Security considerations for MANETs are found in [RFC2501].

The security considerations found in [RFC2529][RFC5214][I-D.nakibly-v6ops-tunnel-loops] also apply to VET.

SEND [RFC3971] and/or IPsec [RFC4301] can be used in environments where attacks on the neighbor coordination protocol are possible. SEAL [I-D.templin-intarea-seal] provides a per-packet identification that can be used to detect source address spoofing.

Rogue neighbor coordination messages with spoofed RLOC source addresses can consume network resources and cause VET nodes to perform extra work. Nonetheless, VET nodes SHOULD NOT "blacklist" such RLOCs, as that may result in a denial of service to the RLOCs' legitimate owners.

VBRs and VBGs observe the recommendations for network ingress filtering [RFC2827].

8. Related Work

Brian Carpenter and Cyndi Jung introduced the concept of intra-site automatic tunneling in [RFC2529]; this concept was later called: "Virtual Ethernet" and investigated by Quang Nguyen under the guidance of Dr. Lixia Zhang. Subsequent works by these authors and their colleagues have motivated a number of foundational concepts on which this work is based.

Telcordia has proposed DHCP-related solutions for MANETs through the CECOM MOSAIC program.

The Naval Research Lab (NRL) Information Technology Division uses DHCP in their MANET research testbeds.

Security concerns pertaining to tunneling mechanisms are discussed in [I-D.ietf-v6ops-tunnel-security-concerns].

Default router and prefix information options for DHCPv6 are discussed in [I-D.droms-dhc-dhcpv6-default-router].

An automated IPv4 prefix delegation mechanism is proposed in [I-D.ietf-dhc-subnet-alloc].

RLOC prefix delegation for enterprise-edge interfaces is discussed in [I-D.clausen-manet-autoconf-recommendations].

MANET link types are discussed in [I-D.clausen-manet-linktype].

The LISP proposal [I-D.ietf-lisp] examines encapsulation/ decapsulation issues and other aspects of tunneling.

Various proposals within the IETF have suggested similar mechanisms.

9. Acknowledgements

The following individuals gave direct and/or indirect input that was essential to the work: Jari Arkko, Teco Boot, Emmanuel Bacelli, Fred Baker, James Bound, Scott Brim, Brian Carpenter, Thomas Clausen, Claudiu Danilov, Chris Dearlove, Remi Despres, Gert Doering, Ralph Droms, Washam Fan, Dino Farinacci, Vince Fuller, Thomas Goff, David Green, Joel Halpern, Bob Hinden, Sascha Hlusiak, Sapumal Jayatissa, Dan Jen, Darrel Lewis, Tony Li, Joe Macker, David Meyer, Gabi Nakibly, Thomas Narten, Pekka Nikander, Dave Oran, Alexandru Petrescu, Mark Smith, John Spence, Jinmei Tatuya, Dave Thaler, Mark Townsley, Ole Troan, Michaela Vanderveen, Robin Whittle, James Woodyatt, Lixia Zhang, and others in the IETF AUTOCONF and MANET

working groups. Many others have provided guidance over the course of many years.

10. Contributors

The following individuals have contributed to this document:

Eric Fleischman (eric.fleischman@boeing.com)
Thomas Henderson (thomas.r.henderson@boeing.com)
Steven Russert (steven.w.russert@boeing.com)
Seung Yi (seung.yi@boeing.com)

Ian Chakeres (ian.chakeres@gmail.com) contributed to earlier versions of the document.

Jim Bound’s foundational work on enterprise networks provided significant guidance for this effort. We mourn his loss and honor his contributions.

11. References

11.1. Normative References

[I-D.templin-intarea-seal] Templin, F., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)", draft-templin-intarea-seal-28 (work in progress), February 2011.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

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

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

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

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

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

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

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

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC5342] Eastlake, D., "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters", BCP 141, RFC 5342, September 2008.

11.2. Informative References

[CATENET] Pouzin, L., "A Proposal for Interconnecting Packet Switching Networks", May 1974.

[I-D.carpenter-flow-ecmp] Carpenter, B. and S. Amante, "Using the IPv6 flow label for equal cost multipath routing and link aggregation in tunnels", draft-carpenter-flow-ecmp-03 (work in progress), October 2010.

[I-D.cheshire-dnsext-multicastdns]
Cheshire, S. and M. Krochmal, "Multicast DNS", draft-cheshire-dnsext-multicastdns-14 (work in progress), February 2011.

[I-D.clausen-manet-autoconf-recommendations]
Clausen, T. and U. Herberg, "MANET Router Configuration Recommendations", draft-clausen-manet-autoconf-recommendations-00 (work in progress), February 2009.

[I-D.clausen-manet-linktype]
Clausen, T., "The MANET Link Type", draft-clausen-manet-linktype-00 (work in progress), October 2008.

[I-D.droms-dhc-dhcpv6-default-router]
Droms, R. and T. Narten, "Default Router and Prefix Advertisement Options for DHCPv6", draft-droms-dhc-dhcpv6-default-router-00 (work in progress), March 2009.

[I-D.ietf-6man-udpzero]
Fairhurst, G. and M. Westerlund, "IPv6 UDP Checksum Considerations", draft-ietf-6man-udpzero-02 (work in progress), October 2010.

[I-D.ietf-dhc-subnet-alloc]
Johnson, R., Kumarasamy, J., Kinnear, K., and M. Stapp, "Subnet Allocation Option", draft-ietf-dhc-subnet-alloc-11 (work in progress), May 2010.

[I-D.ietf-grow-va]
Francis, P., Xu, X., Ballani, H., Jen, D., Raszuk, R., and L. Zhang, "FIB Suppression with Virtual Aggregation", draft-ietf-grow-va-04 (work in progress), February 2011.

[I-D.ietf-lisp]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-10 (work in progress), March 2011.

[I-D.ietf-manet-smf]
Macker, J. and S. Team, "Simplified Multicast Forwarding", draft-ietf-manet-smf-11 (work in progress), March 2011.

[I-D.ietf-v6ops-tunnel-security-concerns]
Krishnan, S., Thaler, D., and J. Hoagland, "Security

Concerns With IP Tunneling",
draft-ietf-v6ops-tunnel-security-concerns-04 (work in progress), October 2010.

[I-D.jen-apt]
Jen, D., Meisel, M., Massey, D., Wang, L., Zhang, B., and L. Zhang, "APT: A Practical Transit Mapping Service", draft-jen-apt-01 (work in progress), November 2007.

[I-D.nakibly-v6ops-tunnel-loops]
Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", draft-nakibly-v6ops-tunnel-loops-03 (work in progress), August 2010.

[IEN48] Cerf, V., "The Catenet Model for Internetworking", July 1978.

[RASADV] Microsoft, "Remote Access Server Advertisement (RASADV) Protocol Specification", October 2008.

[RFC0994] International Organization for Standardization (ISO) and American National Standards Institute (ANSI), "Final text of DIS 8473, Protocol for Providing the Connectionless-mode Network Service", RFC 994, March 1986.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1070] Hagens, R., Hall, N., and M. Rose, "Use of the Internet as a subnetwork for experimentation with the OSI network layer", RFC 1070, February 1989.

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[RFC1753] Chiappa, J., "IPng Technical Requirements Of the Nimrod Routing and Addressing Architecture", RFC 1753, December 1994.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC1955] Hinden, R., "New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG", RFC 1955, June 1996.

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

October 1996.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

[RFC2491] Armitage, G., Schulter, P., Jork, M., and G. Harter, "IPv6 over Non-Broadcast Multiple Access (NBMA) networks", RFC 2491, January 1999.

[RFC2501] Corson, M. and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999.

[RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, March 1999.

[RFC2775] Carpenter, B., "Internet Transparency", RFC 2775, February 2000.

[RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.

[RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, May 2005.

[RFC3947] Kivinen, T., Swander, B., Huttunen, A., and V. Volpe, "Negotiation of NAT-Traversal in the IKE", RFC 3947, January 2005.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

[RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", RFC 4192, September 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

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

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

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

[RFC4548] Gray, E., Rutemiller, J., and G. Swallow, "Internet Code Point (ICP) Assignments for NSAP Addresses", RFC 4548, May 2006.

[RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, June 2006.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[RFC4795] Aboba, B., Thaler, D., and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)", RFC 4795, January 2007.

[RFC4852] Bound, J., Pouffary, Y., Klynsma, S., Chown, T., and D. Green, "IPv6 Enterprise Network Analysis - IP Layer 3 Focus", RFC 4852, April 2007.

[RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, June 2007.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.

[RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, January 2010.

[RFC5685] Devarapalli, V. and K. Weniger, "Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)",

RFC 5685, November 2009.

[RFC5720] Templin, F., "Routing and Addressing in Networks with Global Enterprise Recursion (RANGER)", RFC 5720, February 2010.

[RFC5887] Carpenter, B., Atkinson, R., and H. Flinck, "Renumbering Still Needs Work", RFC 5887, May 2010.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

[RFC6139] Russert, S., Fleischman, E., and F. Templin, "Routing and Addressing in Networks with Global Enterprise Recursion (RANGER) Scenarios", RFC 6139, February 2011.

[RFC6179] Templin, F., "The Internet Routing Overlay Network (IRON)", RFC 6179, March 2011.

Appendix A. Duplicate Address Detection (DAD) Considerations

A priori uniqueness determination (also known as "pre-service DAD") for an RLOC assigned on an enterprise-interior interface would require either flooding the entire enterprise network or somehow discovering a link in the network on which a node that configures a duplicate address is attached and performing a localized DAD exchange on that link. But, the control message overhead for such an enterprise-wide DAD would be substantial and prone to false-negatives due to packet loss and intermittent connectivity. An alternative to pre-service DAD is to autoconfigure pseudo-random RLOCs on enterprise-interior interfaces and employ a passive in-service DAD (e.g., one that monitors routing protocol messages for duplicate assignments).

Pseudo-random IPv6 RLOCs can be generated with mechanisms such as CGAs, IPv6 privacy addresses, etc. with very small probability of collision. Pseudo-random IPv4 RLOCs can be generated through random assignment from a suitably large IPv4 prefix space.

Consistent operational practices can assure uniqueness for VBG-aggregated addresses/prefixes, while statistical properties for pseudo-random address self-generation can assure uniqueness for the RLOCs assigned on an ER’s enterprise-interior interfaces. Still, an RLOC delegation authority should be used when available, while a passive in-service DAD mechanism should be used to detect RLOC duplications when there is no RLOC delegation authority.

Appendix B. Anycast Services

Some of the IPv4 addresses that appear in the Potential Router List may be anycast addresses, i.e., they may be configured on the VET interfaces of multiple VBRs/VBGs. In that case, each VET router interface that configures the same anycast address must exhibit equivalent outward behavior.

Use of an anycast address as the IP destination address of tunneled packets can have subtle interactions with tunnel path MTU and neighbor discovery. For example, if the initial fragments of a fragmented tunneled packet with an anycast IP destination address are routed to different egress tunnel endpoints than the remaining fragments, the multiple endpoints will be left with incomplete reassembly buffers. This issue can be mitigated by ensuring that each egress tunnel endpoint implements a proactive reassembly buffer garbage collection strategy. Additionally, ingress tunnel endpoints that send packets with an anycast IP destination address must use the minimum path MTU for all egress tunnel endpoints that configure the same anycast address as the tunnel MTU. Finally, ingress tunnel endpoints should treat ICMP unreachable messages from a router within the tunnel as at most a weak indication of neighbor unreachability, since the failures may only be transient and a different path to an alternate anycast router quickly selected through reconvergence of the underlying routing protocol.

Use of an anycast address as the IP source address of tunneled packets can lead to more serious issues. For example, when the IP source address of a tunneled packet is anycast, ICMP messages produced by routers within the tunnel might be delivered to different ingress tunnel endpoints than the ones that produced the packets. In that case, functions such as path MTU discovery and neighbor unreachability detection may experience non-deterministic behavior that can lead to communications failures. Additionally, the fragments of multiple tunneled packets produced by multiple ingress tunnel endpoints may be delivered to the same reassembly buffer at a single egress tunnel endpoint. In that case, data corruption may result due to fragment misassociation during reassembly.

In view of these considerations, VBGs that configure an anycast address should also configure one or more unicast addresses from the Potential Router List; they should further accept tunneled packets destined to any of their anycast or unicast addresses, but should send tunneled packets using a unicast address as the source address.

Appendix C. Change Log

(Note to RFC editor - this section to be removed before publication as an RFC.)

Changes from -14 to -15:

- new insights into default route configuration and next-hop determination

Changes from -13 to -14:

- fixed Idnits

Changes from -12 to -13:

- Changed "VGL" *back* to "PRL"
- More changes for multi-protocol support
- Changes to Redirect function

Changes from -11 to -12:

- Major section rearrangement
- Changed "PRL" to "VGL"
- Brought back text that was lost in the -10 to -11 transition

Changes from -10 to -11:

- Major changes with significant simplifications
- Now support stateless PD using 6rd mechanisms
- SEAL Control Message Protocol (SCMP) used instead of ICMPv6
- Multi-protocol support including IPv6, IPv4, OSI/CLNP, etc.

Changes from -09 to -10:

- Changed "enterprise" to "enterprise network" throughout
- dropped "inner IP", since inner layer may be non-IP
- TODO - convert "IPv6 ND" to SEAL SCMP messages so that control messages remain *within* the tunnel interface instead of being

exposed to the inner network layer protocol engine.

Changes from -08 to -09:

- Expanded discussion of encapsulation/decapsulation procedures
- cited IRON

Changes from -07 to -08:

- Specified the approach to global mapping using virtual aggregation and BGP

Changes from -06 to -07:

- reworked redirect function
- created new section on VET interface encapsulation
- clarifications on nexthop selection
- fixed several bugs

Changed from -05 to -06:

- reworked VET interface ND
- anycast clarifications

Changes from -03 to -04:

- security consideration clarifications

Changes from -02 to -03:

- security consideration clarifications
- new PRLNAME for VET is "isatav2.example.com"
- VET now uses SEAL natively
- EBGs can support both legacy ISATAP and VET over the same underlying interfaces.

Changes from -01 to -02:

- Defined CGA and privacy address configuration on VET interfaces
- Interface identifiers added to routing protocol control messages for link-layer multiplexing

Changes from -00 to -01:

- Section 4.1 clarifications on link-local assignment and RLOC autoconfiguration.
- Appendix B clarifications on Weak End System Model

Changes from RFC5558 to -00:

- New appendix on RLOC configuration on VET interfaces.

Author's Address

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707 MC 7L-49
Seattle, WA 98124
USA

Email: fltemplin@acm.org