Internet Draft
Expires: March 2005

Alia Atlas, Ed (Avici Systems)

Loop-Free Alternates for IP/LDP Local Protection

draft-ietf-rtgwg-ipfrr-spec-base-00.txt

Status of this Memo

Abstract

This document defines an architecture and selection process for providing local protection for IP unicast and/or LDP traffic in the event of a single link or node failure until the router has converged. When computing the primary next-hop for a prefix, a router S also determines an alternate next-hop which can be used if the primary next-hop fails. The alternate next-hop is said to be a loop-free alternate, which goes to a neighbor whose shortest path to the prefix does not go back through the router S.

Contents

1. Introduction

Applications for interactive multimedia services such as VoIP and pseudo-wires can be very sensitive to traffic loss, such as occurs when a link or router in the network fails. A router's convergence time is generally on the order of seconds; the application traffic may be sensitive to losses greater than 10s of milliseconds.

As discussed in [FRAMEWORK], minimizing traffic loss requires a mechanism for the router adjacent to a failure rapidly invoke a repair path, which is minimally affected by any subsequent re-convergence. This document describes such a mechanism which allows a

router whose local link has failed to forward traffic to a pre-computed alternate until the router installs the new primary next-hops based upon the changed network topology.

When a local link fails, a router currently must signal the event to its neighbors via the IGP, recompute new primary next-hops for all affected prefixes, and only then install those new primary next-hops into the forwarding plane. Until the new primary next-hops are installed, traffic directed towards the affected prefixes is discarded. This process can take seconds.

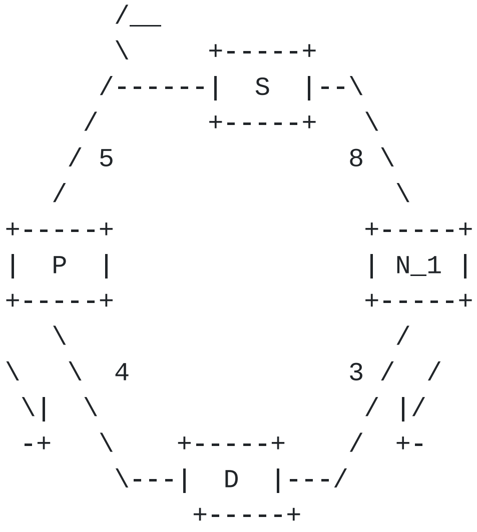

Figure 1: Basic Topology

The goal of IP/LDP Local Protection is to reduce that traffic convergence time to 10s of milliseconds by using a pre-computed alternate interface, in the event that the currently selected primary interface fails, so that the alternate can be rapidly used when the failure is detected.

To clarify the behavior of IP/LDP Local Protection, consider the simple topology in Figure 1. When router S computes its shortest path to router D, router S determines to use the interface to router P as its primary next-hop. Without IP/LDP Local Protection, that interface is the only next-hop that router S computes to reach D. With IP/LDP Local Protection, S also looks for an alternate next-hop interface to use. In this example, S would determine that it could send traffic destined to D by using the interface to router N_1 and therefore S would install the interface to N_1 as its alternate next-hop. At some later time, the link between router S and router P could fail. When that link fails, S (and most likely P) will be the first to detect it. On detecting the failure, S will stop sending traffic destined for D towards P via the failed link, and instead

send the traffic to S's pre-computed alternate next-hop, which is the interface to N_1, until a new SPF is run and its results are installed. As with the primary next-hop, an alternate next-hop is computed for each destination. The process of computing an alternate next-hop does not alter the primary next-hop computed via a standard SPF. The alternate next-hop can protect against a single link or node failure.

If in the example of Figure 1, the link cost from N_1 to D increased to 30 from 3, then N_1 would not be a loop-free alternate, because the cost of the path from N_1 to D via S would be 17 while the cost from N_1 directly to D would be 30. In real networks, we may often face this situation. The existence of a suitable loop-free alternate next-hop is topology dependent.

2. Terminology

SPT --- Shortest Path Tree

D --- The destination router under discussion.

S --- The source router under discussion. It is the viewpoint from which IP/LDP Local Protection is described.

P --- The router which is the primary next-hop neighbor to get from S to D. Where there is an ECMP set for the shortest path from S to D, these will be referred to as P_1, P_2, etc.

N_i --- The ith neighbor of S

R_i_j --- The jth neighbor of N_i, the ith neighbor of S.

Distance_!S(N_i, D) --- The distance of the shortest path from N_i to D which does not go through router S.

Distance_opt(A, B) --- The distance of the shortest path from A to B.

Reverse Distance of a node X --- This is the Distance_opt(X, S).

Loop-Free Alternate --- This is a next-hop that is not a primary next-hop whose shortest path to the destination from the alternate neighbor does not go back through the router S. This is also known as a downstream path or a feasible alternate.

Downstream Path --- This is a loop-free alternate.

Link(A->B) --- A link connecting router A to router B.

```
____\   This is an arrow indicating the primary next-hop towards D.
    /
```

```
@@@@\   This is an arrow indicating the alternate next-hop towards D
    /
```

Primary Neighbor --- One or more of the primary next-hops for S to reach the destination D goes directly to this neighbor.

Loop-Free Neighbor --- A Neighbor N_i which is not the primary neighbor and whose shortest path to D does not go through S.

Loop-Free Node-Protecting Alternate --- This is a path via a Loop-Free Neighbor N_i which does not go through the particular primary neighbor of S which is being protected to reach the destination D.

Loop-Free Link-Protecting Alternate --- This is a path via a Loop-Free Neighbor N_i which does go through the particular primary neighbor of S which is being protected to reach the destination D.

Upstream Forwarding Loop --- This is a forwarding loop which involves a set of routers, none of which are directly connected to the link which has caused the topology change that triggered a new SPF in any of the routers.

3. Finding an Alternate

As with primary next-hops, an alternate next-hop is discussed in relation to a particular destination router D.   For this discussion, the following terminology, as described earlier and illustrated in Figure 2, will be used.

In IP routing, a router S can join the shortest path tree (SPT) at exactly one point -- itself. A loop-free alternate next-hop allows traffic from S to D to deviate from the SPT and then rejoin it. For instance, if S were to send traffic destined for D to N_1 instead of P, thereby deviating from the SPT, then when N_1 received it, N_1 would send that traffic along its shortest path to D.

```
                                           +-----+
                          \          /    _| R_2 |
               +-----+__    \|      |/    / +-----+
               | N_3 |  \   -+     +- __/       \
```

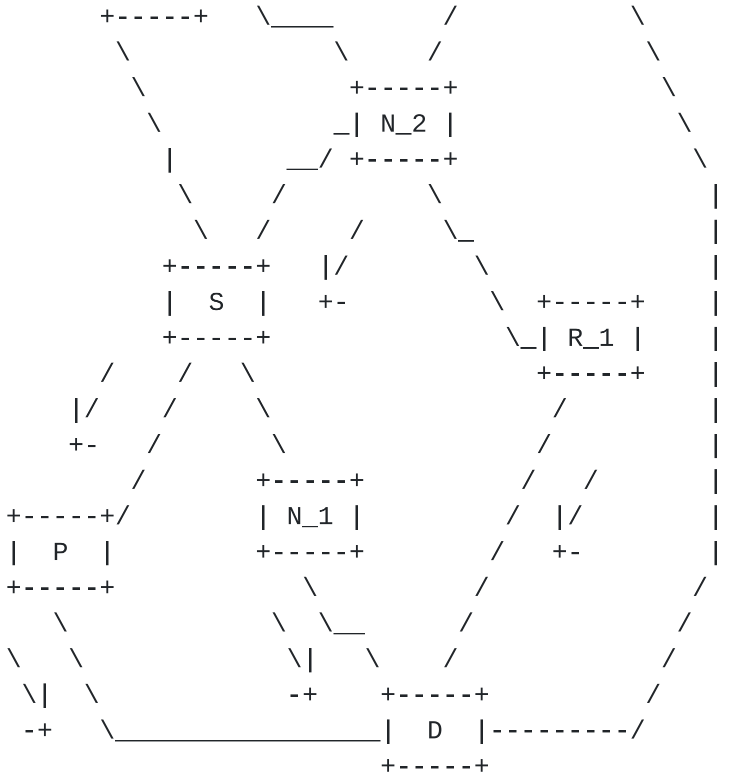

Figure 2: Topology for Terminology

3.1. Loop-Free Alternates

With loop-free alternates, the goal is to expand the set of points at which S can cause its traffic to join the SPT. To illustrate this let's first consider S's neighbors. Router S has the ability to send traffic to any one of its neighbors N_i; this is the easiest possible deviation from the SPT that S can cause to happen. Thus, all of router S's neighbors are candidate alternates at which S could cause traffic to rejoin the SPT. However, it is not useful for router S to use a next-hop which results in traffic rejoining the SPT upstream of S, such that the traffic will transit S again. This would cause a loop. Avoiding a loop is thus the first constraint imposed on the alternate next-hop. In Figure 2, S's neighbors N_2 and N_3 are not loop-free alternate neighbors.

A next-hop which goes to a neighbor that does not have a loop back to S and is not the primary next-hop may be selected as an alternate next-hop. In Figure 2, that is the case for S's neighbor N_1. N_1 is referred to as a loop-free alternate with respect to traffic flowing from S to D because there is no loop caused by forwarding traffic for D to N_1.

An algorithm run on router S must be able to determine which

neighbors provide loop-free alternates. By running an SPF computation from S's perspective, router S can determine the distance from a neighbor N_i to the destination D for the optimal path that does not go through S. This is referred to as Distance_!S(N_i, D). If a neighbor N_i is a loop-free alternate, then it must be cheaper (a lower metric) to get to the destination D without returning to S. This gives the following requirement, where Distance_opt(A, B) gives the distance of the optimal path from A to B.

```
   Distance_!S(N_i, D) < Distance_opt(N_i, S) + Distance_opt(S, D)
```

Equation 1: Criteria for a Loop-Free Alternate

To check this equation, we can consider the other conditions where this is not true. Recall that a router will take the shortest path to a destination that it can see. Thus, if Distance_!S(N_i, D) > Distance_opt(N_i, S) + Distance_opt(S, D), then router N_i will, based on its own shortest path computations, determine to send traffic destined for D to S. Similarly, if Distance_!S(N_i, D) = Distance_opt(N_i, S) + Distance_opt(S, D), then router N_i has equal cost paths to the destination D where one or more of those paths go through S. In such a case where a router N_i has an ECMP set to reach the destination and one or more paths go through S, then the router N_i cannot provide a loop-free alternate because some traffic destined to D may be sent back to S by N_i.

3.2. Selection of an Alternate

The selection of the alternate to use depends upon the failure scenario for which the protection is intended. As with other protection mechanisms, the alternate selected will protect against only a single failure. It is possible to protect against a node failure, which appears as correlated link failures, by explicitly selecting a loop-free alternate which does not use that node.

3.2.1 Failure Scenarios

The simplest case is to locate an alternate which protects against a link failure.

A loop-free link-protecting alternate may cause traffic looping in the event of a node failure. This issue is illustrated in Figure 3. If Link(S->P) fails, then the link-protecting alternate via N will work correctly. However, if router P fails, then both S and N will detect a failure and switch to their alternates. In this example, that would cause S to redirect the traffic to N and N to redirect the traffic to S and thus causing a forwarding loop. Such a scenario can

arise because the key assumption, that all other routers in the network are forwarding based upon the shortest path, is violated because of a second simultaneous correlated failure - another link connected to the same primary neighbor.

```
                                /
                            \   @@@
                          @@@   \
                   +-----+  /    +-----+
                   |  S  |-------|  N  |
                   +-+---+   5   +-----+
                     |              |
                     | 5          5 |  |
                  |  |              | \|/
                 \|/ |              |
                     |    +-----+   |
                     +----|  P  |---+
                          +--+--+
                             |
                             |
                             | 10
                             |
                          +--+--+
                          |  D  |
                          +-----+
```

Figure 3: Link-Protecting Alternates Causing Loop on Node Failure

Such a scenario may be a concern if node failure is not otherwise protected against.

One way to solve such an issue is to add a constraint that the loop-free alternate is loop-free with respect to P and the destination. This gives a loop-free node-protecting alternate. An alternate will be node-protecting if it doesn't go through the same primary neighbor as the primary next-hop. This is the case if Equation 2 is true, where N is the neighbor providing a loop-free alternate.

    Distance_opt(N, D) < Distance_opt(N, P) + Distance_opt(P, D)

However unlike Equation 1, where if the equation did not hold, the neighbor wasn't loop-free, if Equation 2 does not hold, the neighbor may still provide a loop-free alternate that is not node-protecting. In the case of ECMP, the neighbor may even provide a node-protecting loop-free alternate, but S cannot determine this.

It may also be desirable to find an alternate which can protect against other correlated failures. In the general case, these are handled by shared risk link groups (SRLGs) where any links in the network can belong to the SRLG. General SRLGs may add unacceptably to the computational complexity of finding a loop-free alternate.

However, a sub-category of SRLGs is of interest and can be applied only during the selection of an acceptable alternate. This sub-category is to express correlated failures of links which are connected to the same router. For example, if there are multiple logical sub-interfaces on the same physical interface, such as VLANs on an Ethernet interface, if multiple interfaces use the same physical port because of channelization, or if multiple interfaces share a correlated failure because they are on the same line-card. This sub-category of SRLGs will be referred to as local-SRLGs. A local-SRLG has all of its member links with one end connected to the same router. Thus, router S could select a loop-free alternate which does not use a link in the same local-SRLG as the primary next-hop. The local-SRLGs belonging to P can be protected against via node-protection; i.e. picking a loop-free node-protecting alternate.

3.2.2 Broadcast and NBMA Interfaces

The computation for node-protection and link-protection is a bit more complicated for broadcast interfaces. In an SPF computation, a broadcast interface is represented as a pseudo-node with links of 0 cost exiting the pseudo-node. For an alternate to be considered link-protecting, it must avoid the pseudo-node. Thus, a potential alternate which doesn't avoid the next node on the primary path cannot be used as an alternate if the next node on the path is a pseudo-node because the potential alternate would use the link that may fail. Additionally, an alternate which would normally be termed node-protecting because it avoided the next node on the primary path may be only link-protecting. If the alternate avoids the pseudo-node but goes through the next node on the path (i.e. the real neighbor of S), then the alternate is link-protecting; if the alternate avoids not only the pseudo-node but the following node on the primary path, then the alternate is node-protecting.

3.2.3 Interactions with ISIS Overload, RFC 3137 and Costed Out Links

As described in RFC 3137, there are cases where it is desirable not to have a router used as a transit node. For those cases, it is also desirable not to have the router used on an alternate path.

For computing an alternate, a router MUST not consider diverting from the SPF tree along a link whose reverse cost is LSInfinity (for OSPF) or whose router has the overload bit set (for ISIS).

In the case of OSPF, if all links from router S to a neighbor N_i have a reverse cost of LSInfinity, then router S cannot consider using N_i as an alternate.

Similarly in the case of ISIS, if N_i has the overload bit set, then S cannot consider using N_i as an alternate.

This preserves the desired behavior of diverting traffic away from a router which is following RFC 3137 and it also preserves the desired behavior when an operator sets the cost of a link to LSInfinity for maintenance, of not permitting traffic across that link unless there is no other path.

If a link or router which is costed out was the only possible alternate to protect traffic from a particular router S to a particular destination, then there will be no alternate provided for protection.

3.2.4 Characterization of Neighbors

Each neighbor N_i can be categorized as to the type of path it can provide to a particular destination D. Once the primary paths paths have been determined and removed from consideration, each neighbor can be characterized as providing a path in one of the following categories for a particular destination D. It is possible for a neighbor to provide both the primary path and a loop-free link-protecting alternate. The path through the neighbor N_i is either a:

Loop-Free Node-Protecting Alternate - not a primary path and the path avoids both S, one of S's primary neighbors on the path to D and the interface connecting S to that primary neighbor.

Loop-Free Link-Protecting Alternate - not a primary path and the path avoids S and an interface connecting S to one of S's primary neighbors, but goes through that primary neighbor on the path to D. Note that some neighbors of this type may have ECMP paths to reach the destination, where some of those paths are independent of the primary neighbor.

Unavailable - because the path goes through S to reach D, because the interface to reach the neighbor is costed out, etc.

3.2.5 Selection Procedure

Once the neighbors have been categorized, a selection can be made. The selection should maximize the failure cases which can be protected against.

The selection procedure depends on whether S has a single primary neighbor or multiple primary neighbors. A node S is defined to have a single primary neighbor only if there are no equal cost paths that go through any other neighbor; i.e., a node S cannot be considered to have a single primary neighbor just because S does not support ECMP.

If S has a single primary neighbor, then S SHOULD select a loop-free node-protecting alternate, if one is available. If none is available, then S MAY select a loop-free link-protecting alternate.

If S has multiple primary neighbors, then S should select an alternate to protect against the failure of each of the primary next-hops. The loop-free alternate selected should be either one of the other primary next-hops or should provide node-protection.

4. Using an Alternate

If an alternate is available, it is used to redirect traffic when the primary next-hop has failed.

When a local interface failure is detected, traffic that was destined to go out the failed interface must be redirected to the appropriate alternate next-hops. The alternate next-hop is pre-computed to be the most appropriate as mentioned in the selection criteria in the event of the failure scenario being protected against (i.e. link or node failure).

IP/LDP Local Protection does not require any mechanisms for the detection of the failure. The same mechanisms that enable RSVP-TE Fast-Reroute can work here. Because the alternate next-hop is pre-computed, it should be extremely fast to switch traffic to use it, exactly as is the case with RSVP-TE Fast-Reroute.

5. Requirements on LDP Mode

Since LDP traffic will follow the path specified by the IGP, it is also possible for the LDP traffic to follow the loop-free alternates 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 Switched Router (LSR) running LDP must distribute its labels for the FECs it can provide to all its neighbors, regardless of whether or not they are upstream. Additionally, LDP must be acting in liberal label retention mode so that the labels which correspond to interfaces that aren't currently

the primary next-hop are stored. Similarly, LDP should be in downstream unsolicited mode, so that the labels for the FEC are distributed other than along the SPT.

If these requirements are met, then LDP can use the loop-free alternates without requiring any targeted sessions or signaling extensions for this purpose.

6. Routing Aspects

An SPF-like computation is run for each topology, which corresponds to a particular OSPF area or ISIS level. The IGP needs to determine the inheritance of loop-free alternates, as determined for singly advertised routes, to multiply advertised routes, for protocols such as BGP and LDP and for inter-area or inter-level routes. These alternates are provided to LDP and BGP for forwarding purposes only; the alternates are not redistributed in any fashion into other protocols.

The alternate next-hop inheritance is described in the context of inter-area routes, but applies equally well to BGP routes and to routes which are advertised by multiple routers in the IGP area.

6.1 Multiple-Region Routing

Routes in different regions inherit their primary next-hops from the border routers (area border routers (ABRs) or level boundary routers) which offer the shortest path to the destination(s) announcing the route. Similarly, routes must inherit their alternate next-hop and will do so from the same border routers. The shortest path to an inter-region route may be learned from a single border router. In that case, both the primary and the alternate next-hops can be inherited from that border router. Figure 4 illustrates this case where D is reached via ABR1; the primary next-hop for ABR1 is P and the loop-free node-protecting alternate is A1.

The shortest path to an inter-region route may be learned from multiple border routers with at least 2 different primary neighbors, as is illustrated in Figure 5. D is reached via ABR1 and ABR2 with equal cost from S. The primary neighbor to reach ABR1 is P1 and the alternate is A1. The primary neighbor to reach ABR2 is P2 and the alternate is A2. In this case, there are equal-cost primary next-hops to reach D and they can protect each other. In this example, the primary next-hops would be to P1 and P2; if the link to P2 failed, then P1 could be used as an alternate and vice-versa. Thus the alternates can be obtained from the primary next-hops.

```
                            .............
                      ......             ......
                   ...                         ...
                 ..                               ..
               ..   10  +-----+    5    +-----+  5  ..
              .  +------| A1  +---------| R1  |-----+ .
            ..   |      +-----+         +-----+     | .
            .    |                                +-----+  10
           .     |                 +--------------| ABR1|---------+
           .     |                 |      5       +-----+         |
          .  +-----+     5     +---+-+                .           |
          .  |  S  |-----------|  P  |------------+   .         +-----+
           . +-----+           +-----+   10       |   .         |  D  |
           .     |                                |   .         +-----+
            .    |                                |  .             |
            ..   |     +-----+                  +-----+  20        |
              .  +-----| A2  |------------------| ABR2|------------+
               .   10  +-----+    5             +-----+
                ...                               ...
                   ...                         ...
                      ......             ......
                            .............
```

Figure 4: Inter-Region Destination via One Border Router

```
                            ..........
                      ......          ......
                   ...                      ...
                 ..                            ..
               ..   10  +-----+    5    +-----+  ..
              .  +------| A1  +---------| R1  |-----+
            ..   |      +-----+         +-----+     |.
            .    |             +-----+            +-----+  10
           .     | +-----------| P1  |------------| ABR1|---------+
           .     | |       5   +-----+    5       +-----+         |
          .  +-----+                                   .          |
          .  |  S  |---+  5    +-----+   10            .        +-----+
           . +-----+   +-------| P2  |------------+   .         |  D  |
           .     |             +-----+            |   .         +-----+
            .    |                                |  .             |
            ..   |     +-----+                  +-----+  20        |
              .  +-----| A2  |------------------| ABR2|------------+
               .   10  +-----+    5             +-----+
                ...                            ...
                   ...                      ...
                      ......          ......
                            ..........
```

Figure 5: Inter-Region Destination via
Multiple Border Routers and Multiple Primary Neighbors

In the third case, the shortest path to an inter-region route may be learned from multiple border routers but with a single primary neighbor. This is shown in Figure 6, where D can be equally reached from S via ABR1 and ABR2. The alternate next-hop to reach ABR1 is A1 while the alternate to reach ABR2 is A2. It is necessary to select one of the alternates to be inherited.

```
                            .............
                      ......             ......
                   ...                         ...
                 ..                               ..
               ..    5  +-----+   15    +-----+ 20  ..
              .  +------| A1  +---------| R1  |-----+ .
            ..   |      +-----+         +-----+     | .
            .    |                                +-----+  10
           .     |                 +--------------| ABR1|---------+
           .     |                 |      15      +-----+         |
          .  +-----+     5     +---+-+                .           |
          .  |  S  |-----------|  P  |------------+   .         +-----+
           . +-----+           +-----+    5       |   .         |  D  |
           .     |                                |   .         +-----+
            .    |                                |  .             |
            ..   |     +-----+                  +-----+  20        |
              .  +-----| A2  |------------------| ABR2|------------+
               .   10  +-----+   15             +-----+
                ...                               ...
                   ...                         ...
                      ......             ......
                            .............
```

Figure 6: Inter-Region Destination via
Multiple Border Routers but One Primary Neighbor

6.1.1 Inheriting Alternate Next-Hops with One Primary Neighbor

The main question when deciding whether an alternate can be inherited is whether or not that alternate will continue to provide the necessary protection. I.e., will the alternate continue to be usable as an alternate and provide the same link or node protection with respect to the destination that was provided with respect to the border router. The relationships shown in Figure 6 will be used for illustrative purposes, although the topology connecting them may be more general than that shown. The proofs and explanations are provided in Appendix A, but the answer is that the alternate will be usable as an alternate and provide at least the same link or node

protection that was provided with respect to the border router. The alternate next-hop inheritance procedure SHOULD select a loop-free node-protecting alternate, if one is available.

6.1.2 OSPF Inter-Area Routes

In OSPF, each area's links are summarized into a summary LSA, which is announced into an area by an Area Border Router. ABRs announce summary LSAs into the backbone area and inject summary LSAs of the backbone area into other non-backbone areas. A route can be learned via summary LSA from one or more ABRs; such a route will be referred to as a summary route.

The alternate next-hop inheritance for summary routes is as described in Section 6.1.1

6.1.3 OSPF External Routing

Rules of inheritance of alternate next-hops for external routes is the same as for inter-area destinations. The additional complication comes from forwarding addresses, where an ASBR uses a forwarding address to indicate to all routers in the Autonomous System to use the specified address instead of going through the ASBR. When a forwarding address has been indicated, all routers in the topology calculate the shortest path to the link specified in the external LSA. In this case, the alternate next-hop of the forwarding link should be used, in conjunction with the primary next-hop of the forwarding link, instead of those associated with the ASBR.

6.1.4 ISIS Multi-Level Routing

ISIS maintains separate databases for each level with which it is dealing. Nodes in one level do not have any information about state of nodes and edges of the other level. ISIS level boundary points , also known as ISIS level boundary routers, are attached to both levels. ISIS level boundary routers summarize the destinations in each, level. ISIS inter-level route computation is very similar to OSPF inter area routing. Rules for alternate next-hop inheritance is the same as described in Section 6.1.1

6.2 OSPF Virtual Links

OSPF virtual links are used to connect two disjoint backbone areas using a transit area. A virtual link is configured at the border routers of the disjoint area. There are two scenarios, depending

upon the position of the root, router S.

If router S is itself an ABR or one of the endpoints of the disjoint area, then router S must resolve its paths to the destination on the other side of the disjoint area by using the summary links in the transit area and using the closest ABR summarizing them into the transit area. This means that the data path may diverge from the virtual neighbor's control path. An ABR's primary and alternate next-hops are calculated by RAPID on the transit area.

The primary next-hops to use are determined based upon the closest set of equidistant ABRs; the same rules described in Section 6.1.1 for inter-area destinations must be followed for OSPF virtual links to determine the alternate next-hop. The same ECMP cases apply.

If router S is not an ABR, then all the destinations on the other side of the disjoint area will inherit the virtual link's endpoint, the transit ABR. The same OSPF inter-area rules described in Section 6.1.1 must be followed here as well.

A virtual link cannot be used as an alternate next-hop.

6.3 BGP Next-Hop Synchronization

Typically BGP prefixes are advertised with AS exit routers router-id, and AS exit routers are reached by means of IGP routes. BGP resolves its advertised next-hop to the immediate next-hop by potential recursive lookups in the routing database. IP/LDP Local Protection computes the alternate next-hops to the all the IGP destinations, which includes alternate next-hops to the AS exit router's router-id. BGP simply inherits the alternate next-hop from IGP. The BGP decision process is unaltered; BGP continue to use the IGP optimal distance to find the nearest exit router. MBGP routes do not need to copy the alternate next hops.

6.4 Multicast Considerations

IP/LDP Local Protection does not apply to multicast traffic. The alternate next-hops SHOULD not used for multi-cast RPF checks.

7. Security Considerations

This document does not introduce any new security issues. The mechanisms described in this document depend upon the network topology distributed via an IGP, such as OSPF or ISIS. It is dependent upon the security associated with those protocols.

8. Full Copyright Statement

9. References

[FRAMEWORK] M. Shand, "IP Fast Reroute Framework", draft-ietf-rtgwg-ipfrr-framework-01.txt, June 2004

[LDP] L. Anderson, P. Doolan, N. Feldman, A. Fredette, B. Thomas, "LDP Specification", RFC 3036, January 2001

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

[RSVP-TE FRR] P. Pan, D. Gan, G. Swallow, JP Vasseur, D. Cooper, A. Atlas, and M. Jork, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", work-in-progress draft-ietf-mpls-rsvp-lsp-fastreroute-06.txt, June 2004

[RFC3137] Retana, A., Nguyen, L., White, R., Zinin, A., and McPherson, D., "OSPF Stub Router Advertisement", RFC 3137, June 2001

[RFC3277] D. McPherson, "Intermediate System to Intermediate System (IS-IS) Transient Blackhole Avoidance", RFC 3277, April 2002

[ISIS] R. Callon, "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments", RFC 1195, December 1990

[RFC2966] T. Li, T. Przygienda, H. Smit, "Domain-wide Prefix Distribution with Two-Level IS-IS", RFC 2966, October 2000

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

[RFC2370] R. Coltun, "The OSPF Opaque LSA Option", RFC 2370, July 1998

10. Authors Information

Raveendra Torvi
Avici Systems
101 Billerica Avenue
N. Billerica, MA 01862
USA
email: rtorvi@avici.com
phone: +1 978 964 2026

Gagan Choudhury
AT&T
Room D5-3C21
200 Laurel Avenue
Middletown, NJ 07748
USA
email: gchoudhury@att.com
phone: +1 732 420-3721

Christian Martin
Verizon
1880 Campus Commons Drive
Reston, VA 20191
email: cmartin@verizon.com

Brent Imhoff
WilTel Communications
3180 Rider Trail South
Bridgeton, MO 63045
USA
email: brent.imhoff@wcg.com
phone: +1 314 595 6853

Don Fedyk

Nortel Networks
600 Technology Park
Billerica, MA 01821
email: dwfedyk@nortelnetworks.com
phone: +1 978 288 3041

11. Editor's Information

Alia Atlas
Avici Systems
101 Billerica Avenue
N. Billerica, MA 01862
USA
email: aatlas@avici.com
phone: +1 978 964 2070

Appendix A: Loop-Free Alternate Proofs

Consider where A2 is a loop-free alternate with respect to S and ABR2. Will A2 be a loop-free alternate with respect to S and D? Let there be three ABRs which must be considered. Each ABR can represent a group of ABRs with the same characteristics.

```
                            .............
                      ......             ......
                   ...                         ...
                 ..                               ..
               ..    5  +-----+   15    +-----+ 20  ..
              .  +------| A1  +---------| R1  |-----+ .
            ..   |      +-----+         +-----+     | .
            .    |                                +-----+  10
           .     |                 +--------------| ABR1|---------+
           .     |                 |      15      +-----+         |
          .  +-----+     5     +---+-+                .           |
          .  |  S  |-----------|  P  |------------+   .         +-----+
           . +-----+           +-----+    5       |   .         |  D  |
           .     |                                |   .         +-----+
            .    |                                |  .             | |
            ..   |     +-----+                  +-----+  20        | |
              .  +-----| A2  |------------------| ABR2|------------+ |
               .   10  +-----+     15           +-----+              |
                ...       |                        ...               |
                   ...    +---------------+    ...                   |
                      ......     10    +--+--+.           20         |
                            ...........| ABRt|------------------------+
                                       +-----+
```

Figure 7: Inter-Region Destination via
Multiple Border Routers but One Primary Neighbor

ABR1 is from the set of ABRs where D_opt(A2, ABR1) = D_opt(A2, S) + D_opt(S, ABR1). In other words, A2 is not loop-free with regards to S and ABR1. Additionally, D_opt(S, D) = D_opt(S, ABR1) + D_opt(ABR1, D) so ABR1 is on a shortest path from S to D.

ABR2 is from the set of ABRs where D_opt(A2, ABR2) < D_opt(A2, S) + D_opt(S, ABR2). In other words, A2 is loop-free with regards to S and ABR2. Additionally, D_opt(S, D) = D_opt(S, ABR2) + D_opt(ABR2, D) so ABR2 is on a shortest path from S to D.

ABRt is from a set of ABRs where D_opt(S, D) < D_opt(S, ABRt) + D_opt(ABRt, D). In other words, ABRt is not on a shortest path from S to D.

First, we will prove that D_opt(A2, D) < D_opt(A2, ABR1) + D_opt(ABR1, D). In other words, the shortest path from A2 to D does not go through ABR1.

The shortest path from A2 to D via ABR1 also goes via S. A shortest path from S to D goes via ABR1.

```
          Step i: D_opt(A2, ABR1) + D_opt(ABR1, D) =
         D_opt(A2, S) + D_opt(S, ABR1) + D_opt(ABR1, D)
```

The shortest path from A2 to D via ABR2 does not go through S. ABR2 is on a shortest path from S to D.

```
          Step ii: D_opt(A2, ABR2) + D_opt(ABR2, D) <
        D_opt(A2, S) + D_opt(S, ABR2) + D_opt(ABR2, D)
```

From previous and given that ABR1 and ABR2 provide equal-cost paths from S to D:

```
         Step iii: D_opt(A2, ABR2) + D_opt(ABR2, D) <
        D_opt(A2, S) + D_opt(S, ABR1) + D_opt(ABR1, D)
```

From previous and Step i:

```
          Step iv: D_opt(A2, ABR2) + D_opt(ABR2, D) <
                D_opt(A2, ABR1) + D_opt(ABR1, D)
```

```
  Step v: D_opt(A2, D) <= D_opt(A2, ABR2) + D_opt(ABR2, D) <
               D_opt(A2, ABR1) + D_opt(ABR1, D)
```

Thus, the optimal path from A2 to D cannot go through ABR1.

Next, we will prove that if D_opt(A2, D) = D_opt(A2, ABRt) +

D_opt(ABRt, D), then A2 is still loop-free with respect to S and D. In other words, even if A2's shortest path to D goes through an ABRt which isn't on a shortest path from S to D, the path from A2 to D is still loop-free with respect to S and D. This is proved via contradiction.

Assume that D_opt(A2, D) goes through ABRt.

Step i: D_opt(A2, ABRt) + D_opt(ABRt, D) <= D_opt(A2, ABR2) + D_opt(ABR2, D)

Because A2 is loop-free with respect to S and ABR2
Step ii: D_opt(A2, ABR2) + D_opt(ABR2, D) < D_opt(A2, S) + D_opt(S, ABR2) + D_opt(ABR2, D)

Because ABR2 is on a shortest path from S to D and ABRt is not
Step iii: D_opt(S, ABR2) + D_opt(ABR2,D) < D_opt(S, ABRt) + D_opt(ABRt, D)

From previous by adding Dopt(A2, S) to both sides
Step iv: D_opt(A2, S) + D_opt(S, ABR2) + D_opt(ABR2,D) < D_opt(A2, S) + D_opt(S, ABRt) + D_opt(ABRt, D)

From Steps i and ii:
Step v: D_opt(A2, ABRt) + D_opt(ABRt, D) < D_opt(A2, S) + D_opt(S, ABR2) + D_opt(ABR2, D)

From Steps iv and v:
Step vi: D_opt(A2, ABRt) + D_opt(ABRt, D) < D_opt(A2, S) + D_opt(S, ABRt) + D_opt(ABRt, D)

Therefore, if D_opt(A2, D) is via ABRt, it does not go through S.

These two proofs show that if A2 is loop-free with respect to S and ABR2, then A2 is loop-free with respect to S and D.

Appendix A.1 Loop-Free Node-Protecting Alternate Proofs

It must also be shown that if A2 is loop-free and node-protecting with respect to S and ABR2, then A2 will still be node-protecting with respect to S and D. In other words, that A2 will be loop-free with respect to P and D.

This is shown where D_opt(S, D) = D_opt(S, P) + D_opt(P, D), so that D_opt(P, ABR1) + D_opt(ABR1, D) = D_opt(P, ABR2) + D_opt(ABR2, D).

First, it has already been proven that an ABR offering equal-cost from S to D which is also loop-free with respect to S and D will be selected by A2 over an ABR offering equal-cost from S to D which is not loop-free with respect to S and D. Since the alternate inheritance is of interest only where all the ABRs offering equal-cost paths to D have the same primary next-hop P, if A2 is loop-free and node-protecting for one ABR offering equal-cost paths to D, then A2 is node-protecting for all those ABRs.

Next, given that A2's optimal path to ABR2 does not go through P, is to prove that if A2's optimal path to D goes via some ABRt, that that path does not go through P. This can be shown using variable replacement of the second proof given as follows:

```
     Assume that D_opt(A2, D) goes through ABRt.
               Step i: D_opt(A2, ABRt) + D_opt(ABRt, D) <=
                     D_opt(A2, ABR2) + D_opt(ABR2, D)

               Step ii: D_opt(A2, ABR2) + D_opt(ABR2, D) <
              D_opt(A2, P) + D_opt(P, ABR2) + D_opt(ABR2, D)

               Step iii: D_opt(P, ABR2) + D_opt(ABR2,D) <
                     D_opt(P, ABRt) + D_opt(ABRt, D)

     From previous by adding Dopt(A2, P) to both sides
        Step iv: D_opt(A2, P) + D_opt(P, ABR2) + D_opt(ABR2,D) <
              D_opt(A2, P) + D_opt(P, ABRt) + D_opt(ABRt, D)

     From Steps i and ii:
               Step v: D_opt(A2, ABRt) + D_opt(ABRt, D) <
              D_opt(A2, P) + D_opt(P, ABR2) + D_opt(ABR2, D)

     From Steps iv and v:
               Step vi: D_opt(A2, ABRt) + D_opt(ABRt, D) <
              D_opt(A2, P) + D_opt(P, ABRt) + D_opt(ABRt, D)

     Therefore, if Dopt(A2, D) is via ABRt, it does not go through P.
```

This proves that if A2 provides a loop-free node-protecting alternate for S to reach ABR2, then A2 will also provide a loop-free node-protecting alternate for S to reach D.