Network Working Group
Internet-Draft
Expires: August 5, 2006

A. Atlas, Ed.
Google, Inc.
February 2006

U-turn Alternates for IP/LDP Fast-Reroute
draft-atlas-ip-local-protect-uturn-03

Status of this Memo

Copyright Notice

Abstract

This document defines and describes the use of U-turn alternates to provide local protection for IP unicast and/or LDP traffic in the event of a single failure, whether link, node or shared risk link group (SRLG). When a topology change occurs, a router S pre-computes for each prefix an alternate next-hop that can be used if the primary next-hop fails. An acceptable alternate can be either a loop-free alternate or a U-turn alternate. A U-turn alternate uses a neighbor, whose primary next-hop to the prefix is router S itself and which has itself a loop-free node-protecting alternate, which thus does not go

through router S to reach the destination prefix.

Table of Contents

1. Introduction

This document extends IP Fast-Reroute, as defined in [I-D.ietf-rtgwg-ipfrr-spec-base] and [I-D.ietf-rtgwg-ipfrr-framework], which allows a router whose local interface or next-hop 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.

The existence of suitable loop-free alternate next-hops is topology dependent. This document defines a second type of alternate next-hop, known as a U-turn alternate, and provides the common behavior and selection method necessary to allow U-turn alternates to function.

The topology in Figure 1 is an example where there is no loop-free alternate from S to D, but there is a U-turn alternate.

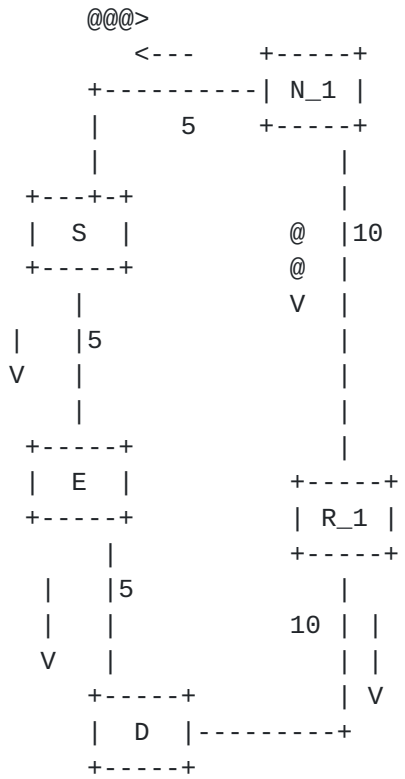

Figure 1: Topology with U-turn Alternate

In Figure 1, there is no loop-free alternate for S to use to reach D. This is because the costs are such that N_1 uses S as its primary neighbor; therefore if S were to send the traffic to N_1, it would loop back to S. If both S and N_1 support the mechanisms defined in this document, then S could use N_1 as a U-turn alternate. Traffic

destined to D that was sent by S to N_1 would be forwarded by N_1 to R_1, N_1's loop-free node-protecting alternate.

In examining realistic networks, it was seen that loop-free alternates did not provide adequate coverage for the traffic between all the source-destination pairs. As with loop-free alternates, the existence of suitable U-turn alternates is topology dependent; it is seen to substantially extend the coverage on realistic topology above that seen with just loop-free alternates.

This document describes the case where a loop-free node-protecting alternate must be available at a neighbor's neighbor. It is possible to extend the length of the U-turn to provide better coverage at the cost of additional local computation.

1.1 Terminology

This document uses the terminology defined in [I-D.ietf-rtgwg-ipfrr-framework] and the additional terms defined below.

Distance_opt(A, B) or D_opt(A,B): The distance of a shortest path from A to B.

Distance_!S(A, B) or D_!S(A,B): The distance of a shortest path from A to B that does not traverse S.

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

U-Turn Alternate: This is an alternate next-hop of S that goes to a neighbor N_i, whose primary next-hop is S, and whose alternate neighbor does not go back trough the router S, which may therefore use the link to N_i as an alternate.

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

---> An arrow indicating the primary next-hop towards D.

@@@> An arrow indicating the alternate next-hop towards D.

U-Turn Neighbor: A neighbor N_i is a U-Turn neighbor of router S with respect to a given destination D if and only if S is a primary neighbor of N_i to reach the destination D for all optimal paths which go through S to reach D.

ECMP U-Turn Neighbor: A neighbor N_i that is a U-Turn neighbor and that has at least one equal cost path to reach D that does not go through S as well as at least one equal cost path that does go through S to reach D.

Looping Neighbor: A neighbor N_i is a looping neighbor of router S with respect to a given destination D if and only if S is not the primary next-hop of N_i on at least one optimal path from N_i to D that also goes through S.

2. U-turn Alternates

As with primary next-hops, an alternate next-hop is discussed in relation to a particular destination router D. As described in [I-D.ietf-rtgwg-ipfrr-spec-base], a neighbor can provide a loop-free alternate if Equation 1 is true.

D_opt(N_i, D) < D_opt(N_i, S) + D_opt(S, D)

Equation 1: Criteria for a Loop-Free Alternate

When there are no loop-free alternates, this means that all of S's remaining non-primary neighbors will send traffic for D back to S, either directly or indirectly. It is probable that one of S's non-primary neighbors will have a loop-free node-protecting alternate that could be utilized if the neighbor N_i is able to identify a packet as a U-turn packet.

N_i can indicate its ability to correctly identify incoming U-turn packets on each layer-3 interface; such an interface is U-turn Recipient capable[ISIS-LOCAL-PROTECT][OSPF-LOCAL-PROTECT]. U-turn packets are identified implicitly or explicitly as described in Section 2.3.

N_i MUST only send U-turn packets to N_i's loop-free node-protecting alternate if the packet is received from a primary neighbor for that destination. This motivates the definitions below of a Looping Neighbor and a U-turn Neighbor. These examples are illustrated in Figure 2.

Looping Neighbor: A neighbor N_i is a looping neighbor of router S with respect to a given destination D if any of N_i's shortest paths to D goes through S but S is not the primary next-hop of N_i for all those paths through S.

U-Turn Neighbor: A neighbor N_i is a U-Turn Neighbor of router S with respect to a given destination D if and only if S is a primary next-hop of N_i to reach the destination D for all optimal paths that go through S to reach D.

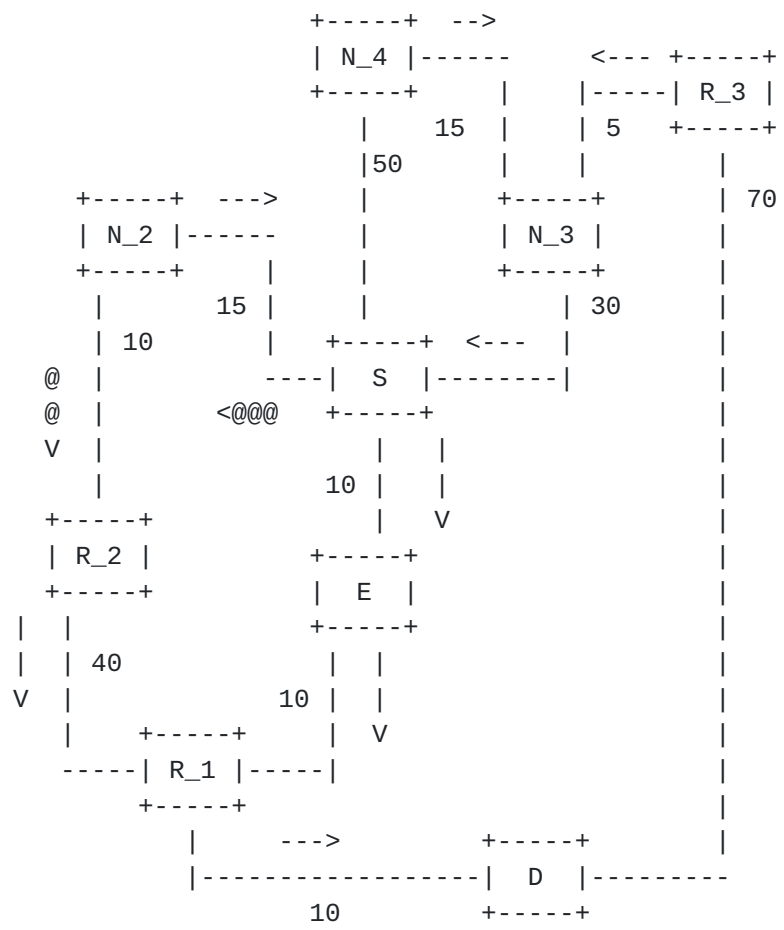

Figure 2: Terminology of Looping Neighbors and Example U-Turn Alternate

Mathematically, for a neighbor N_i to be a U-Turn neighbor, it is necessary that Equation 1 be false. If D_opt(N_i,D) = D_opt(N_i,S) + D_opt(S,D), then there may be multiple optimal paths, at least one of which goes through S and one does not. If the shortest distance path from N_i to D that doesn't traverse S (D_!S(N_i, D)) is equal to D_opt(N_i, S) + D_opt(S, D, then there are multiple optimal paths where at least one traverses S and one does not. Such a neighbor may be an ECMP U-Turn neighbor or may be a looping neighbor.

Additionally, all optimal paths to reach D that go via S must be via a direct link between N_i and S. If a neighbor N_i does not satisfy Equation 1 and all optimal paths to reach D that go via S are via a direct link between N_i and S, then it is a U-turn neighbor.

2.1 ECMP U-turn Neighbors

The above definition for U-Turn Neighbor allows a neighbor, which has equal cost paths (an ECMP set) where at least one of those paths goes directly to S and others may or may not, to be a U-Turn Neighbor. Consider the topology shown in Figure 3. In this figure, N_1 has three equal-cost paths to reach D which are N_1 - S - E - D, N_1 - R_1 - D, and N_1 - R_2 - D. Because the only path that goes through S goes directly through S, N_1 is a U-Turn neighbor of S. A neighbor is an ECMP U-turn neighbor if an optimal path from N_i to D traverses S and there are multiple optimal paths from N_i to D.

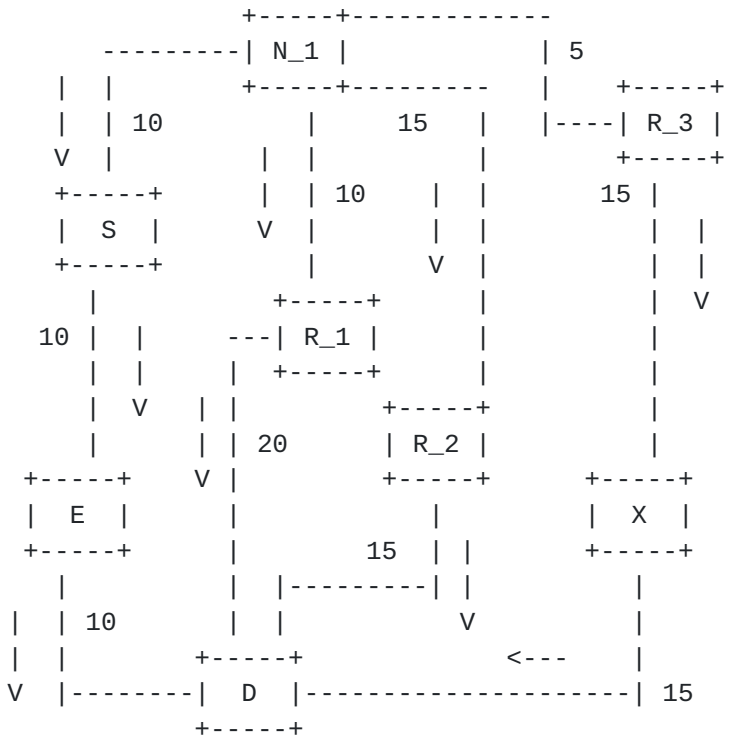

Figure 3: ECMP U-Turn Neighbor

S does not know whether a neighbor N_i supports ECMP or how that neighbor selects among the equal cost paths. Recall that a node will only direct U-turn packets to the alternate if those packets are received from that node's primary neighbors.

Consider the topology in Figure 4, where N_2 has three equal cost primary neighbors which are S, N_1 and R_1. If N_2 were to select only N_1 as its primary neighbor, then N_2 would break U-Turns only on traffic received from N_1 and not on traffic received from S. Therefore, S cannot consider N_2 as an ECMP U-Turn neighbor because S cannot rely upon N_2 to break U-turns for traffic destined to D which

is received from S.

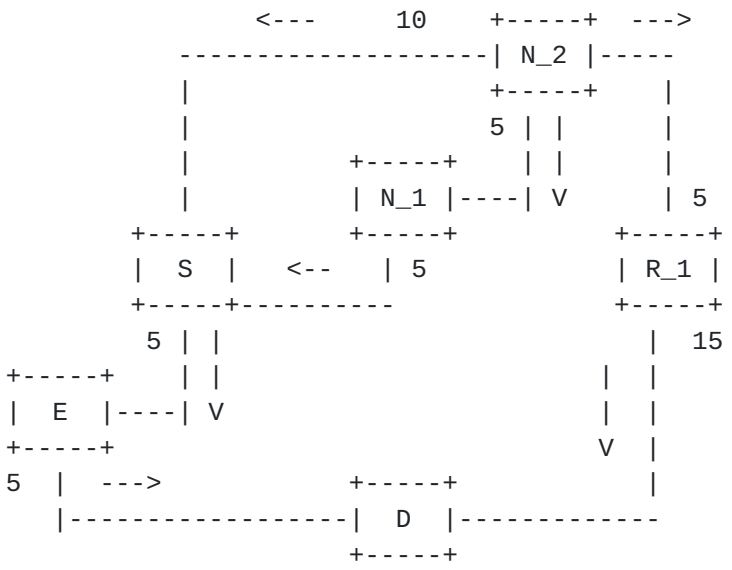

Figure 4: ECMP Neighbor Which is Not an ECMP U-Turn Neighbor

If N_2 has multiple paths to reach D that traverse S and not all such paths have S as the next-hop, then S cannot use N_2 as a U-Turn neighbor.

2.2 U-turn Neighbor's Alternate

For router S to use a U-turn neighbor N_i for a U-turn alternate, N_i requires a loop-free node-protecting alternate[I-D.ietf-rtgwg-ipfrr-spec-base]. If R_i_j provides a loop-free node-protecting alternate for N_i and S is N_i's primary neighbor, then the path from R_i_j to D will not traverse S. The requirement for an R_i_j to provide a suitable alternate is:

$$D\_opt(R\_i\_j, D) < D\_opt(R\_i\_j, S) + D\_opt(S, D)$$

Equation 2: Loop-Free Node-Protecting Neighbor's Neighbor

Because N_i is a U-turn neighbor, N_i's shortest path to D traverse S; therefore Equation 2 means that the shortest paths from R_i_j to D also do not traverse N_i.

Each router independently computes the alternate that it will select for a given destination D. For the U-turn alternate to provide broadcast link protection, or node or SRLG protection, the router N_i must consistently select a suitable alternate, if available, such

that N_i's primary neighbor S can determine what R_i_j is providing that alternate.

2.3 Identifying U-turn Traffic

There are two methods for identifying a packet as a U-turn packet. The methods are implicit U-turn packet identification and explicit U-turn packet identification. These methods are described in Section 2.3.1 and Section 2.3.2.

A router supporting this specification MUST support one of these two methods for identifying U-turn packets. A

2.3.1 Implicit U-turn Packet Identification

The first method requires no modification to the packets sent into the U-turn alternate. In this method, when, on an Implicit U-turn Recipient Capable interface or sub-interface, a packet for a destination D is received from the primary neighbor to which the router would forward the packet, then the router locally identifies the packet as a U-turn packet and forwards on the loop-free alternate. In essence, this breaks the single hop micro-forwarding loop.

2.3.1.1 Broadcast and NBMA Interfaces

NBMA and broadcast interfaces can be treated identically for IP/LDP Fast-Reroute; both involve the case of possibly receiving traffic from multiple neighbors. With broadcast links (i.e. Gigabit Ethernet), there can be multiple neighbors connected to the same link. If all the neighbors are not in the same IGP area or there are hosts with default routes on the link, then explicit U-turn packet identification MUST be used.

If implicit U-turn packet identification were used, traffic could be incorrectly sent to the alternate. It is extremely desirable to have at most one forwarding table per interface or sub-interface. If all the neighbors are in the same IGP area, it still MUST be considered whether all traffic received on an interface can be treated identically, regardless of the neighbor sourcing the traffic on that interface; otherwise explicit packet identification SHOULD be used.

The cost for any node on the broadcast link to reach S or its primary neighbor E will be identical. Because all link costs are positive, no neighbor on the broadcast link will ever send traffic to S along that interface in order to reach E. Therefore, S can assume that any traffic received from the broadcast link that goes to a destination via a primary next-hop neighbor that is also on the broadcast link is

in fact sent by that primary next-hop neighbor and should be redirected to break the U-Turn.

Thus, if router S has a primary next-hop neighbor for a given prefix on the broadcast link, S should redirect all traffic received destined to that prefix on the broadcast link to S's alternate next-hop.

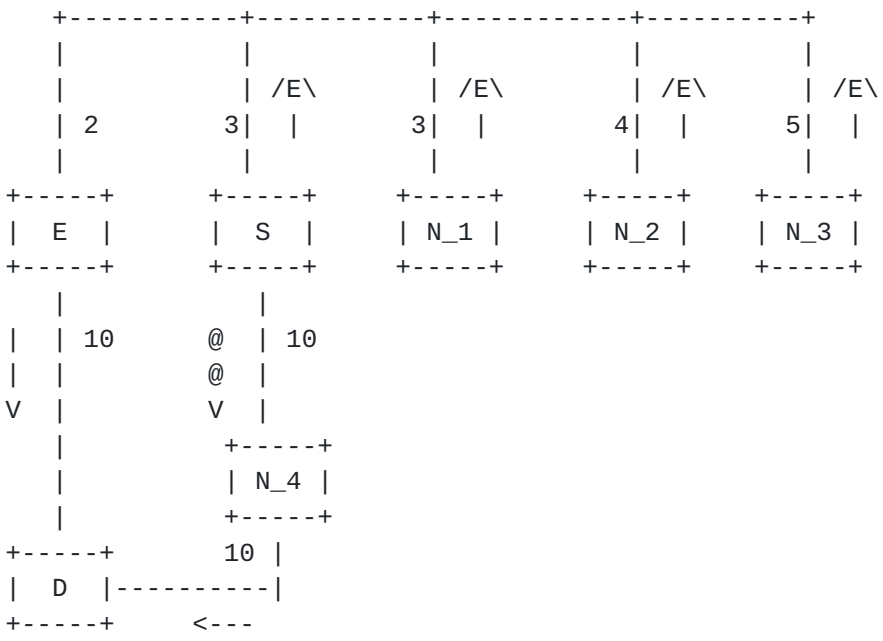

Figure 5: Topology With Broadcast Link

If it were acceptable to have one forwarding table per neighbor on the link, then the restriction that all neighbors on a broadcast link be in the same IGP region and not be hosts with default routes could be removed.

2.3.2 Explicitly Marked U-turn Packet Identification

The second method requires that U-turn packets be explicitly marked as such by the router that is directing the packet into the U-turn alternate. This method is motivated by the following benefits:

a. For certain existing hardware platforms, it may be difficult to implicitly detect packets as coming from a primary neighbor and forward those packets differently. An explicit marking permits straightforward U-turn handling.

b. For broadcast and NBMA links, if packets in the U-turn alternate are not explicitly marked, there are restrictions on the

neighbors and hosts (see Section 2.3.1.1). This could limit realistic deployment scenarios where hosts may exist on the same broadcast link as routers. When U-turn packets are explicitly marked, the router can treat some packets received on the interface as U-turn packets and some as normal packets. This permits routers and hosts on a link to send normal traffic while the primary neighbor can send explicitly marked U-turn packets.

c. If a router were to request penultimate-hop popping (PHP) for an LSP egressing on interface that the router had also advertised as U-turn Recipient capable, then it would be possible for traffic exiting that LSP to be mis-identified using the implicit identification. If U-turn packets are explicitly marked, then this confusion would not occur and the router could both request PHP for LSPs egressing an interface and supported explicit U-turn packet identification.

Explicitly marking U-turn traffic has the following disadvantages, which could be viewed as advantages for the implicit U-turn traffic:

a. A marking method must be selected. This marking will need to be below Layer 3; there are certainly no available bits for this purpose in the IPv4 header.

b. In some cases implicit U-turn marking will mitigate loops that form by detecting the loop and forwarding to a loop-free node-protecting alternate. This capability is lost when packets are explicitly marked.

There are a number of different ways in which U-turn packets could be explicitly marked. For example, this could be done at Layer 2 by using different PPP types, Ethernet types, etc. The simplest mechanism that can apply regardless of the Layer 2 technology is to use a well-known MPLS label (referred to as a U-turn Label). By requiring that routers supporting this specification use the same well-known MPLS label, there is no need to communicate the label.

There are already different PPP types, Ethernet types, etc. for MPLS. If a router does not support any other MPLS mechanism, then a packet received with the U-turn label can be clearly identified from the layer-2 information indicating that the packet is MPLS. The MPLS label on the packet SHOULD be checked to verify that the label is the U-turn label.

Unlike the common use of MPLS labels, the U-turn label does not indicate specifically where the packet should be switched. The U-turn label indicates that the packet should be tentatively identified as a U-turn packet. The label is always popped on the

receiving node.

If the explicitly marked packet was received from a primary neighbor, then the packet is a U-turn packet; the U-turn label MUST be popped and the decision on where to forward the packet is based on the packet's identification as a U-turn packet and the packet's destination IP address or the new top MPLS label (after the U-turn label has been popped).

If the explicitly marked packet was not received from a primary neighbor, then the packet is not a U-turn packet, the U-turn label must be popped, and the packet MUST be forwarded as a normal packet based upon its destination IP address or the top MPLS label (after the U-turn label has been popped). This scenario could occur if a failure happened during another topology change where the sending router will be or was the receiving router's primary neighbor.

It is always necessary to check whether a U-turn marked packet was received from a primary neighbor and to determine from which primary neighbor to properly handle cases where the receiving router has equal-cost paths to the destination. For example, in Figure 3 N_1 has three equal-cost paths via S, R_1 and R_2. Assume that N_1 has selected S and R_1 as its primary next-hops. When N_1 receives a U-turn marked packet from S, then that packet can be sent to R_1. When N_1 received a U-turn marked packet from R_1, then that packet can be sent to S. When N_1 receives a U-turn marked packet from R_2, N_1 determines it didn't come from a primary neighbor and will send it to either S or R_1. The need to determine which primary neighbor a U-turn marked packet came from can be seen even more clearly if, for this example, N_1 had selected only S as its primary next-hop and selected R_1 as the loop-free node-protecting alternate next-hop. N_1 might receive U-turn marked packets from S, R_1 or R_2; N_1 must not forward the packets received from R_1 back to R_1.

The QoS characteristics associated with a packet with a U-turn label SHOULD be based on the IP packet or the MPLS packet after the U-turn label has been removed.

3. Example Algorithm for finding U-turn Alternates

This section describes an algorithm that allows the identification of U-turn alternates with a single reverse-SPF computation rooted at S and at most one additional SPF computation per neighbor that could be used as a U-turn alternate. These are required in addition to those required to locate loop-free alternates.

The computational complexity of locating alternates is extremely important. There are several factors which potentially influence

this.

N: Number of neighbors of S

A: Number of neighbors that could be used as alternates

U: Number of neighbors that could be used as U-turn alternates

Clearly, any path computation mechanisms will depend upon the cost of SPF calculations, which depend upon the number of links and nodes and pseudo-nodes in the network and the parameter above. However, different approaches can lead to very different numbers of SPF calculations, ranging from a number of computations proportional to N (or A and U) up to a number proportional to the number of nodes in the network, the number of local links, the number of neighbors' neighbors, or even the number of differently homed prefixes. Clearly, the latter are undesirable.

A single SPF is done to find the primary next-hops; this yields D_opt(S, D) for all D. The additional computation required for loop-free alternates is at worst an SPF rooted at each neighbor N_i that can be used as an alternate. This gives a worst-case of an additional A SPF computations to find loop-free alternates. The information obtained is D_opt(D, S) and D_opt(N_i, D) for all N_i and D.

It is important to understand the minimum computation required for U-turn alternates beyond that needed for loop-free alternates. The first information required is the distance from any neighbor's neighbor R_i_j back to S; this is known via a single reverse SPF rooted at S. The minimum information that must be determined is whether a particular neighbor N_i has a loop-free node-protecting alternate. This can be determined for a neighbor N_i by running a single U-turn SPF. To explain the rationale behind the mechanisms in a U-turn SPF, consider the following.

An SPF algorithm is performing a minimization across the potential paths. A regular SPF is started by exploring each link connected to the root N_i and using the metric associated with that link as the cost. Therefore, at each destination D, it determines D_opt(N_i, D).

If instead each link from the root N_i is explored with a cost of 0, then, if there are J neighbors of N_i, the distance associated with the path at each destination D would be
min_forall j in J (D_!N_i(R_i_j, D))
where D_!N_i indicates the shortest path from the particular R_i_j to D that does not traverse N_i.

Now, if one considers the loop-free test from Equation 2 and groups all the R_i_j terms onto one side, one obtains the following equation:

D_opt(R_i_j, D) - D_opt(R_i_j, S) < D_opt(S, D).

If an SPF could be modified to minimize the left-hand side of the above equation for all R_i_j neighboring N_i, then the resulting value could be compared to D_opt(S,D) to determine if N_i had a loop-free node-protecting alternate. Mathematically, if there are 1 to J different neighbors of N_i, the desired result at each destination D would be:
min_forall j in J (D_opt(R_i,j, D) - D_opt(R_i_j, S)).

It is sufficient to determine at each destination D:

min_forall j in J (D_!N_i(R_i,j, D) - D_opt(R_i_j, S)).

Equation 4: Path Minimization for U-turn Alternate Check

To visualize this, consider the following 2 different cases where N_i's primary neighbor is S.

A shortest path from R_i_j to D is via N_i and thus S. Therefore, D_!N_i(R_i_j, D) >= D_opt(R_i_j, D).

A shortest path from R_i_j to D is not via N_i. Therefore, D_!N_i(R_i_j, D) = D_opt(R_i_j, D).

Now that the rationale behind a U-turn SPF is clearer, here is the description of a U-turn SPF. If this procedure is followed, then the stored path distance at each destination D will be Equation 3.

A U-turn SPF is a regular SPF where the initial exploration of links from the root N_i uses different costs depending upon the node at the other end of the link. Links from N_i to a node R_i_j are explored with a cost of -D_opt(R_i_j, S). If a link goes from N_i to a pseudo-node, then the pseudo-node's links are also explored as part of this step. The pseudo-node itself is not given a non-infinite path distance in this step. In this step, each link from a pseudo-node neighboring N_i to a node R_i_j is explored with a cost of -D_opt(R_i_j, S). At the end of this step, each R_i_j will be on the candidate-list. From this point, the normal mechanics of the Dijkstra algorithm apply; when a node is removed from the candidate-list, its links will be explored with the cost that of the link metric.

Links from N_i will not be explored if those links are not available

to provide alternate protection. First, if a point-to-point link is connected to S, it is not explored. Second, a link to a pseudo-node, where that pseudo-node is also connected to S, with the cost of D_opt(N_i, S) will not be explored; it is possible that an alternate found via that link would not provide link-protection for N_i's primary next-hop. Third, a link that is configured to not be used as an alternate will not be explored. Fourth, a link whose forward or reverse cost is at the maximum will not be explored; such a cost indicates that it desired for the link not to be used to tranist traffic.

To support ECMP U-turn alternates, it is necessary to know the path traversal without going through S. Therefore, in the U-turn SPF computation, S is never placed onto the candidate list; its links are never explored.

From the above description of a U-turn SPF and the rationale behind it, it can be seen that at most one U-turn SPF is needed per neighbor that could be used as a U-turn alternate. The computational complexity of a U-turn SPF is roughly the same as a regular SPF. The additional computational complexity is U U-turn SPF computations, where U is the number of neighbors that could be considered as U-turn alternates.

The above gives the ability to determine if a neighbor N_i has a loop-free node-protecting alternate and can therefore provide a U-turn alternate. It does not provide a method to determine if that U-turn alternate is node-protecting. Because D_opt(R_i_j, E) is not known as a result of the previous SPFs, a simple distance comparison is not possible without additional SPFs. To obtain D_opt(R_i_j, E) would require R SPF computations and replace the U U-turn SPF computations. In a network the number of neigbhors is generally much less then the number of neighbors' neighbors. Therefore, another method of determining node protection for U-turn alternates is desirable.

During the U-turn SPF, it is possible to track those neighbors of S that were visited along the stored path. If this is done and N_i always selects the R_i_j corresponding to that path as an alternate, then S can determine whether that stored path traverses E, S's primary next-hop. This allows S to determine node-protection at the cost of a bit of additional book-keeping. A similar method is required to determine link protection for broadcast links; the neighboring pseudo-nodes must be tracked.

This discussion of an algorithm to compute U-turn alternates is intended to provide explanatory background for the selection procedure defined in Section 4.5.1.

3.1 SRLG Protection

It may be desirable to obtain an alternate that provides SRLG protection. If SRLGs are being considered, then this would need to be signaled to neighboring routers; an extension to the router capability would provide the mechanism.

It can be determined if a U-turn alternate provides SRLG-protection by tracking the SRLGs traversed. This may miss a possible U-turn alternate; to locate all possible U-turn alternates and determine SRLG protection may need an SPF computation per neighbors' neighbor. Intelligent pruning of the R_i_j considered based upon first link SRLGs may improve the completeness of the algorithm while not requiring an SPF computation per neighbors' neighbor.

4. Alternate Next-Hop Calculation

A router S must select an appropriate alternate next-hop. A common selection method is required to support U-turn alternates. At a minimum, a router must select a loop-free node-protecting alternate. It is necessary for router S to know the path taken by the R_i_j that will be selected by N_i; if multiple R_i_j might be used, then the paths are combined.

The same set of failure scenarios that can be protected against with a loop-free alternate is of interest with a U-turn alternate. Just as downstream paths solve concerns with micro-forwarding loops via alternates in the event of unplanned for failures, the same concerns can be solved for U-turn alternates by ensuring that the selected neighbor's neighbor is a downstream path with respect to S (D_opt(R_i_j, D) < D_opt(S, D)).

There is also the same interaction with maximum costed links and broadcast interfaces as described in [I-D.ietf-rtgwg-ipfrr-spec-base]. In addition, if all links from a neighbor N_i to a neighbor's neighbor R_i_j have a cost or reverse maximum cost (LSInfinity for OSPF), then router S cannot consider that N_i could provide a U-turn alternate via R_i_j. The rationales for these restrictions are the same as given in [I-D.ietf-rtgwg-ipfrr-spec-base].

4.1 IP/LDP Fast-Reroute Alternate Capability

There are a number of different reasons why an operator may not wish for a particular interface to be used as an alternate. For instance, the interface may go to an edge router or the interface may not have sufficient bandwidth to contain the traffic which would be put on it in the event of failure.

If an interface should not be used as an alternate, then the router MUST signal this appropriately (e.g. as specified in [I-D.ietf-isis-link-attr] and in [OSPF-LOCAL-PROTECT]) to indicate "link excluded from local protection path". The neighbor routers must not consider that such links might be capable of providing a loop-free node-protecting alternate. Therefore, this "link excluded from local protection path" capability is flooded as part of the link capabilities information. Links that are not capable of being alternates are not explored in the first step of the U-turn SPF.

Because a router's neighbors may desire to use that router to provide a U-turn alternate, a router must flood to its neighbors which interfaces are not capable of providing alternates. This information allows a router's neighbors to accurately determine whether or not the router has a loop-free node-protecting alternate.

4.2 U-turn Recipient Capabilities

A router S can only use a neighbor as a U-turn alternate next-hop if that neighbor has advertised its ability to identify U-turn alternates on a link to S. The implicit U-turn recipient capability and/or the explicit marked U-turn recipient capability must be signaled by a neighbor for a link in order for S to determine that it is allowed to use that neighbor as a potential U-turn alternate. By default, S MUST assume that a neighbor cannot provide a U-Turn alternate unless that neighbor indicates the implicit or the explicit marked U-Turn recipient capability on the link to be used by S to reach that neighbor.

The U-turn alternate next-hop MUST use a link which has been advertised as implicit or explicit marked U-turn Recipient capable by the intended neighbor. If router S is only capable of sending unmarked U-turn packets, then router S MUST not use links which are not advertised as implicit U-turn Recipient capable to reach a U-turn alternate next-hop. Similarly, if router S is only capable of sending marked U-turn packets, then router S MUST not use links which are not advertised as explicit marked U-turn Recipient capable to reach a U-turn alternate next-hop.

If a link is advertised only as explicit marked U-turn Recipient capable and it is selected to reach the U-turn alternate next-hop, then router S MUST apply the marking, as described in the explicit marked U-turn packet identification method, to each packet sent into the U-turn alternate. If the link is advertised only as implicit U-turn Recipient capable and it is selected to reach the U-turn alternate next-hop, then router S MUST not apply any additional marking. If the link is advertised as both implicit U-turn Recipient capable and explicit marked U-turn Recipient capable, then router S

may make a local decision as to whether to apply the additional marking.

The extensions to signal the U-turn recipient capability and the Marked U-turn recipient capability are described in [OSPF-LOCAL-PROTECT] and [ISIS-LOCAL-PROTECT].

4.3 Link-Protecting U-turn Alternate

For a neighbor N_i to be useable by a router S as a U-turn alternate next-hop to reach a destination D to protect against a link between S and a primary next-hop E, the following topology-based conditions MUST be true.

1. D_opt(N_i, D) >= D_opt(N_i, S) + D_opt(S, D)

2. N_i is either a U-turn neighbor or an ECMP U-turn neighbor. In other words, S is always the primary next-hop on all shortest paths from N_i to D that traverse S.

3. N_i has a loop-free link-protecting node-protecting alternate (as computed in the U-turn SPF):
   min_forall j in J (D_!N_i(R_i,j, D) - D_opt(R_i_j, S)) < D_opt(S, D)

4. The path traversed in the U-turn SPF at N_i didn't traverse the pseudo-node form S to E

5. N_i can be reached via a candidate alternate next-hop that doesn't traverse the link from S to E or any pseudo-node along that link.

6. If N_i is an ECMP U-turn neighbor, then all other equal-cost paths must be loop-free with respect to link from S to E:
   D_!S(N_i, D) < D_!S(N_i, pseudo_S_E) + D_opt(pseudo_S_E, D)

If N_i is an ECMP U-turn neighbor, S cannot determine whether N_i has selected S as a primary neighbor. Therefore, N_i must both pass the tests for a U-turn neighbor while ignoring the equal-cost paths from N_i that don't go through S and the tests for a loop-free neighbor while ignoring the equal-cost paths from N_i that do go through S. More specifically, the loop-free conditions are verified using D_!S instead of D_opt; the U-turn conditions are verified by looking at the path traversal.

The non-topology based conditions are dependent upon the signaled link capabilities as described earlier.

4.4 U-turn Node-Protecting Alternate

For a U-turn alternate next-hop to protect against node failure, S must be able to determine the set of R_i_j that might be used to provide the loop-free node-protecting alternate to N_i. All optimal paths from each of those R_i_j to the destination D MUST avoid S's primary neighbor E. This is expressed by the following topology-based conditions that MUST be true.

1. D_opt(N_i, D) >= D_opt(N_i, S) + D_opt(S, D)

2. N_i is either a U-turn neighbor or an ECMP U-turn neighbor. In other words, S is always the primary next-hop on all shortest paths from N_i to D that traverse S.

3. N_i has a loop-free link-protecting node-protecting alternate (as computed in the U-turn SPF):
   min_forall j in J (D_!N_i(R_i,j, D) - D_opt(R_i_j, S)) < D_opt(S, D)

4. The path traversed in the U-turn SPF at N_i didn't traverse E

5. If N_i is an ECMP U-turn neighbor, then all other equal-cost paths must be loop-free with respect to E:
   D_!S(N_i, D) < D_!S(N_i, E) + D_opt(E, D)

For a U-turn alternate to be both link-protecting and node-protecting, it must meet the requirements in this section and in Section 4.3.

4.5 Selection Procedure

A router supporting this specification SHOULD select a loop-free alternate next-hop or a U-turn alternate next-hop for each primary next-hop used for a given prefix. If a router advertised either the explicit or implicit U-turn recipient capability on any link, then the router MUST select a loop-free node-protecting link-protecting alternate next-hop for each primary next-hop used for a given prefix, if such an alternate is available. A router MAY decide to not use an available loop-free or U-turn alternate next-hop. The selection should maximize the failure cases that can be protected against.

A router MAY use different alternate(s) for forwarding U-turn packets and for forwarding traffic when a primary next-hop fails. The alternate(s) used when a primary next-hop fails are a router-local decision.

A router S can only be used as a U-turn alternate next-hop by its

primary neighbor E if S selects a loop-free link-protecting node-protecting alternate next-hop. Therefore a router MUST select a loop-free link-protecting node-protecting alternate if one is available. Otherwise, a router MAY select any other type of available alternate.

A candidate alternate next-hop may be connected to a primary neighbor, a loop-free neighor, a U-turn neighbor, and ECMP U-turn neighbor or a looping neighbor. The heirarchy among the alternate next-hops is as follows, with the first listed the most preferred.

1. Next-hop to a primary neighbor with link and node protection.

2. Next-hop to a primary neighbor with at least link protection.

3. Next-hop to a loop-free neighbor with link and node protection.

4. Next-hop that offers some level of protection.

The protection provided by a next-hop that connects to a primary neighbor can be determined in the same way as the protection provided by a next-hop that connects to a loop-free neighbor. These conditions are given in [I-D.ietf-rtgwg-ipfrr-spec-base], but for clarity are briefly repeated below.

a. Loop-free for S:
   D_opt(N_i, D) < D_opt(N_i, S) + D_opt(S, D)

b. Path Loop-free for link from S to E:
   D_opt(N_i, D) < D_opt(N_i, pseudo_S_E) + D_opt(pseudo_S_E, D)

c. Candidate next-hop doesn't use the pseudo-node from S to E, if any.

d. Loop-free for E:
   D_opt(N_i, D) < D_opt(N_i, E) + D_opt(E, D)

The following describes the alternate selection for a particular primary next-hop to a destination. Initially no alternate next-hop is selected. Each candidate alternate next-hop is considered in turn and either replaces the alternate next-hop or is removed from consideration. This description assumes that a single alternate next-hop is selected; it is possible to have a set of alternate next-hops. In that case, all members MUST be from a set where it is a router-local decision on how to decide among them.

If the candidate connects to a primary neighbor and provides link and node protection, then the candidate MUST replace any alternate next-

hop lower in the heirarchy. How to handle ties is a router-local decision. If the candidate connects to a primary neighbor and provides only link protection, then the candidate MAY replace any alternate next-hop lower in the heirarchy.

If the candidate connects to a loop-free neighbor and provides link protection and node protection, then if the alternate next-hop is not higher on the heirarchy, a decision as to whether to replace the alternate next-hop with the candidate MUST be made as described in Section 4.5.1.

Any other type of candidate alternate next-hop MUST NOT replace an alternate next-hop that is higher in the heirarchy. Beyond this restriction, the decision among such candidates is router-local.

4.5.1 Selection Between Multiple Loop-Free Node-Protecting Alternate

The specific selection policy described in this section is motivated by the ability to reduce the computational complexity associated with identifying U-turn alternates. This mechanism is explained in Section 3.

D_opt(R_i_j, D) - D_opt(R_i_j, S)

Equation 5: Shortest Reverse-Path-Discounted Distance from R_i_j to D

A consequence of this mechanism is that the only path traced during the U-turn SPF is that of the shortest reverse-path-discounted path. A second consequence is that the optimal distance between a neighbor's neighbor and S's primary neighbor E (D_opt(R_i_j, E)) is not always known.

By constraining the loop-free node-protecting alternate selection as specified below, it is sufficient to know only the path of the shortest reverse-path-discounted path via any of N_i's neighbors.

The selection by a router among loop-free link-protecting node-protecting alternates MUST be as follows.

Each loop-free node-protecting alternate next-hop is a specific R_i_k where there are K members. The selected R_i_m must be provide the shortest reverse-path-discounted path among all the R_i_j.

D_opt(R_i_m, D) - D_opt(R_i_m, S) =
min_forall k in K (D_opt(R_i_k, D) - D_opt(R_i_k, S)

If there are multiple such R_i_m and one provides the destination, then that one SHOULD be selected. Otherwise, if there are multiple

such R_i_m, the router SHOULD make a consistent selection.

A consequence of this selection algorithm is that, all else being equal, a more expensive link from an R_i_j will be preferred. This should be considered when determining appropriate metrics for IP traffic-engineering.

5. Using an Alternate

If an alternate is available, it may be used in two circumstances. The first is when a local failure is detected; the behavior on a local failure is that specified in [I-D.ietf-rtgwg-ipfrr-spec-base]. The second is when U-turn packets are received and the alternate is loop-free and node-protecting.

5.1 Alternate Use On Failure

If an alternate next-hop is available, the router SHOULD redirect traffic to the alternate next-hop when the primary interface has failed. If the alternate next-hop provides node protection, the router SHOULD redirect traffic to the alternate next-hop when the primary next-hop has failed and the detection of that failure has occurred within an appropriately short period. The mechanisms available for failure detection are discussed in [I-D.ietf-rtgwg-ipfrr-framework] and are outside the scope of this specification.

The alternate next-hop MUST be used only for traffic types which are routed according to the shortest path. Multicast traffic is specifically out of scope for this specification.

The details in [I-D.ietf-rtgwg-ipfrr-spec-base] on terminating the use of the alternate apply equally to U-turn alternates.

Although extremely unlikely in examined topologies, it is theoretically possible that the convergence on the part of the U-turn neighbor N_i could cause a short micro-forwarding loop as in the following topology.

In this example, N provides a U-turn alternate to S via the loop-free node-protecting alternate A. After the link from S to E fails, N's alternate continues to function. When N converges, the new primary next-hop is B; if B has not already converged, then a micro-loop between N and B could form.

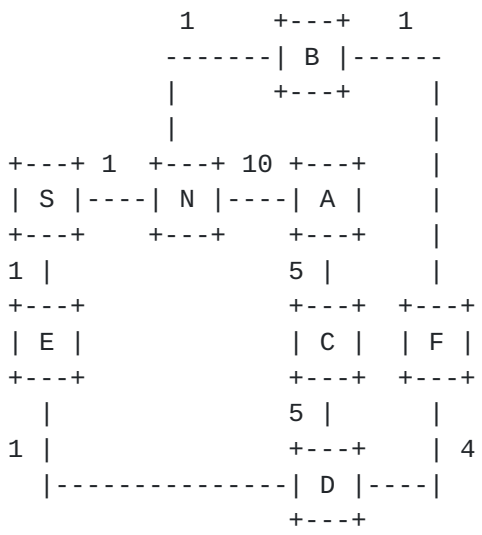

Figure 6: Micro-loop Affecting U-turn Alternate

To avoid such an unlikely circumstance, a router SHOULD delay installation of the new primary and alternate next-hops for a destination if the failed link is connected to a primary neighbor and there is a loop-free node-protecting alternate to protect that primary neighbor and that alternate was not a shortest path to D (before the failure).

This installation delay SHOULD terminate

a. if the new primary next-hop was loop-free prior to the topology change, or

b. if a configured hold-down, which represents a worst-case bound on the length of the network convergence transition, has expired, or

c. if notification of an unrelated topological change in the network is received.

This delay is required only due to the possibility that the U-turn alternate next-hop may have a new primary neighbor that was not loop-free prior to the failure. The loop-free node-protecting alternate of N_i which goes via R_i,j will not be affected by the failure, because it was independent of the affected elements. If N_i's new primary neighbor remains S, then the traffic will continue to be directed towards the appropriate R_i,j. If N_i converges to a path that does not include S to reach D, then traffic received from S for D will be sent along the new path and a micro-forwarding loop is theoretically possible.

5.2 U-turn Packets Forwarding

If a packet is received from a primary neighbor and is successfully identified as a U-turn packet (see Section 2.3), then a router which supports this specification MUST send the packet to the loop-free node-protecting alternate, selected according to the rules in this specification, that is associated with the primary next-hop to that neighbor. If, on a U-turn Recipient capable interface, a packet is received from a non-primary neighbor (who believes that it is a primary neighbor) and the packet is marked to indicate that it is a U-turn packet, then a router which supports this specification MUST send the packet to a primary next-hop.

6. LDP Interactions and Routing Aspects

6.1 LDP Interactions

U-turn alternates do not impose any additional sessions or signaling on LDP. LDP can use the U-turn alternates to protect LDP traffic if the requirements specified in [I-D.ietf-rtgwg-ipfrr-spec-base] are met.

6.2 Multi-Homed Prefixes

The treatment of multi-homed prefixes is the same as with loop-free alternates [I-D.ietf-rtgwg-ipfrr-spec-base]. A multi-homed prefix p can be treated in the SPF computations as a node with uni-directional links to it from those routers that have advertised the prefix.

If a router is advertising the ability of at least one link to be implicit or explicit U-turn recipient capable, then a router MUST compute the alternate next-hop for an IGP multi-homed prefix by considering alternate paths via all routers that have announced that prefix.

6.3 OSPF

There are some applicability restrictions for OSPF in regard to loop-free alternates. Similar ones will apply for U-turn alternates. Additional restrictions may apply and more details will be available in the next revision.

6.4 U-turn Alternates Interactions with Tunnels

IP Fast-Reroute treats IGP tunnels the same as any other link. If router S is not the endpoint of the tunnel, then the alternate path is computed as normal. If router S is the ingress into the tunnel, then all destinations, which have the tunnel as a primary next-hop,

may be protected either via a protection scheme associated with the tunnel or via IP FRR.

One issue with MPLS RSVP-TE tunnels is that an LSP may be created where the router uses penultimate-hop popping (PHP). If the implicit U-turn packet identification method is used, then traffic received via that tunnel is undistinguishable from traffic received over the interface. If some packets received via the LSP are destined back to the penultimate hop, then the egress router would consider that those were U-turn packets and redirect that traffic to its alternate, if available. To avoid such a scenario, a router can simply not request PHP for those LSPs which are entering via an interface upon which the router has advertised that it can break U-Turns. Alternately, a router could use the explicit U-turn packet identification method. If that is not supported and the router must do PHP, then the router can stop advertising the link as U-turn recipient capable.

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. Acknowledgements

The authors would like to thank Joel Halpern for his helpful review and comments, particularly as pertains to Section 3.

9. Intellectual Property Considerations

Avici Systems has intellectual property rights claimed in regard to the specification contained in this document.

10. References

[I-D.ietf-isis-link-attr]
Vasseur, J. and S. Previdi, "Definition of an IS-IS Link Attribute sub-TLV", draft-ietf-isis-link-attr-01 (work in progress), May 2005.

[I-D.ietf-rtgwg-ipfrr-framework]
Shand, M. and S. Bryant, "IP Fast Reroute Framework", draft-ietf-rtgwg-ipfrr-framework-05 (work in progress), March 2006.

[I-D.ietf-rtgwg-ipfrr-spec-base]
Atlas, A. and A. Zinin, Ed., "Basic Specification for IP

Fast-Reroute: Loop-free Alternates", draft-ietf-rtgwg-ipfrr-spec-base-05.txt (work in progress), February 2006.

[ISIS-LOCAL-PROTECT] Atlas, A., Torvi, R., and C. Martin, "ISIS Extensions to support U-turn Alternates for IP/LDP Fast-Reroute", draft-martin-isis-local-protect-cap-02.txt (work in progress), February 2006.

[OSPF-LOCAL-PROTECT] Atlas, A., Torvi, R., Choudhury, G., Martin, C., Imhoff, B., and D. Fedyk, "OSPFv2 Extensions for Link Capabilities to support U-turn Alternates for IP/LDP Fast-Reroute", draft-atlas-ospf-local-protect-cap-02.txt (work in progress), February 2006.

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

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

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

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

[RFC3036] Andersson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", RFC 3036, January 2001.

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

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

Authors' Addresses

Alia K. Atlas (editor)
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
USA

Email: akatlas@alum.mit.edu

Raveendra Torvi
Avici Systems, Inc.
101 Billerica Avenue
N. Billerica, MA 01862
USA

Phone: +1 978 964 2026
Email: rtorvi@avici.com

Gagan Choudhury
AT&T
200 Laurel Avenue, Room D5-3C21
Middletown, NJ 07748
USA

Phone: +1 732 420-3721
Email: gchoudhury@att.com

Christian Martin
iPath Technologies

Email: chris@ipath.net

Brent Imhoff
Juniper Networks
1194 North Mathilda
Sunnyvale, CA 94089
USA

Phone: +1 314 378 2571
Email: bimhoff@planetspork.com

Don Fedyk
Nortel Networks
600 Technology Park
Billerica, MA 01821
USA

Phone: +1 978 288 3041
Email: dwfedyk@nortelnetworks.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.