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

This Internet-Draft will expire on November 6, 2012.

Copyright Notice

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.

```
            +---+             +---+
            | B |-------------| C |
            +---+             +---+
           /                       \
          /                         \
         /                           \
    +---+            +---+  failure   +---+
    |Src|------------| A |-----X------|Dst|
    +---+            +---+            +---+
      =========>=============>=========>
                  Primary path
```

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:

```
Legend: X : period with packet loss
        FN forwarding delay: |--|

      |--|--------|
A:----1XX2XXXXXXXX3--------------------------------------------------
         |----|   |----|
B:------------4--5-----6XX7------------------------------------------
              |--|--------|

(a) Link delay is |----| FIB update delay is |--------|



      |--|--------|
A:----1XX2XXXXXXXX3--------------------------------------------------
         |---------------|
                  |---------------|
B:-----------------------4--5-----6XX7-------------------------------
                         |--|--------|

(b) Link delay is |---------------| FIB update delay is |--------|
```

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:

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. 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

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           FN Length           |  FN App Type  | AuType|unused |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Originator Router ID                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Neighbour Router ID                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Link ID                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Sequence Number                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Sequence Number (cont'd)                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

```
      FIB lookup       |         FIB lookup
                       |
ASBR2 ========> NH2    |   ASBR2 ========> NH2 <----+
ASBR1 ========> NH1    |   ASBR1 ========> NH1 <-+  |
                       |                         |  |
P1    ========> NH1    |   P1    ========> Ptr1 -+  |
P2    ========> NH1    |   P2    ========> Ptr1 -+  |
P3    ========> NH1    |   P3    ========> Ptr1 -+  |
P4    ========> NH1    |   P4    ========> Ptr1 -+  |
                       |                            |
P5    ========> NH2    |   P5    ========> Ptr2 ----+
P6    ========> NH2    |   P6    ========> Ptr2 ----+
P7    ========> NH2    |   P7    ========> Ptr2 ----+
                         |
```

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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               16              |     0x008     | AuType|unused |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         ABR Router ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        ASBR Router ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               0x0                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Sequence Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Sequence Number (cont'd)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

```
 +---+   +---+   +---+   +---+
 | E |---| F |---| G |---| H |
 +---+   +---+   +---+   +---+
   |            /          |
   |           /           |
   |          /            |
 +---+   +---+   +---+   +---+
 | A |---| B |---| C |-X-| D |
 +---+   +---+   +---+   +---+
   >========>==============>
      Traffic from A to D
```

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.

```
+-------+-------+-------+--   --+-------+
|   r1  |   r2  |   r3  |  ...  |   rk  |
+-------+-------+-------+--   --+-------+
    |       |       |               |
    |       |       |               |
   \|/     \|/     \|/             \|/
    *       *       *               *
+-------+-------+-------+--   --+-------+
| fail1 | fail2 | fail3 |       | failk |
| alt.1 | alt.2 | alt.3 |  ...  | alt.k |
+-------+-------+-------+--   --+-------+
| fail4 |       | fail5 |
| alt.4 |       | alt.5 |
+-------+       +-------+
| fail6 |
| alt.6 |
+-------+
```

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	20	3.64	82.39	98.85	100.0
beta=0.01	50	3.86	82.10	98.69	100.0
	100	3.98	83.21	98.04	100.0
p(add)=0.05	20	3.70	85.60	99.14	100.0
beta=0.05	50	4.01	84.17	99.09	100.0
	100	4.08	83.35	98.01	100.0
p(add)=0.1	20	5.52	93.24	100.0	100.0
beta=0.15	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	154 (AT&T)	6.88	91.04	99.81	100.0
beta=0.62	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