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

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

August 20, 2013

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

Abstract

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Enabling fast dissemination of a network event to routers in a limited area could benefit multiple applications. Existing use cases are centered around new approaches for IP Fast ReRoute such as [I-D.csaszar-ipfrr-fn]. In the future, however, multiple innovative applications may take advantage of a Fast Notification service.

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

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

1.1. Requirements Language

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

1.2. Acronyms

FN	-	Fast Notification
IGP	-	Interior Gateway Protocol
IS-IS	-	Intermediate System to Intermediate System
MD5	-	Message Digest 5
OSPF	-	Open Shortest Path First
RPF	-	Reverse Path Forwarding
SHA	-	Secure Hash
SPT	-	Shortest Path Tree
STP	-	Spanning Tree Protocol

2. Design Goals

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

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

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

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

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

5. The mechanism should have low processing overhead.

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

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

3. Transport Logic - Distribution of the Notifications

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

3.1. Flooding mode

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

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

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

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

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

3.1.1. Duplicate Check with Flooding

Duplicate check can be performed in numeruous ways.

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

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

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

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

3.2. Spanning Tree Mode

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

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

4. Message Encoding

4.1. Seamless Encapsulation

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

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

4.2. Dedicated FN Message

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

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

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      +-                                                             -+
      |                           IP Header                           |
      +-                +-------------+                              -+
      |                 | Protocol=UDP|                               |
      +-                +-------------+                              -+
      |                                                               |
      +-                                                             -+
      |                                                               |
      +---------------------------------------------------------------+
      |     UDP Source Port = FN      |   UDP Destination Port = FN   |
      +---------------------------------------------------------------+
      |                       UDP Header cont’d                       |
      +---------------------------------------------------------------+
      |                           FN Header                           |
      +---------------------------------------------------------------+
      |                              ...                              |
      .                                                               .
      .                          FN Payload                           .
      .                                                               .
      |                              ...                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                              ...                              |
      .                                                               .
      .                     Authentication (optional)                 .
      .                                                               .
      |                              ...                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: FN packet format as a UDP datagram

The encoding of the FN Header is as follows:

```
       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 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: FN Header encoding

FN Length (16 bits)

The length of the FN message in bytes including the FN Header and the FN Payload. The authentication data optionally appended to the FN packet is not considered part of the FN message: the authentication data is not included in the FN Length field, although it is included in the length field of the packet’s IP header.

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

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

4.2.1. Authentication

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

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

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   AuLength    |        ... Authentication Data ...            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

Figure 3: Authentication field in FN packets

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

The authentication type may be manually pre-configured or may be selected automatically. For automatic selection, the nodes have to know what type of authentication is applicable for the rest of the nodes. This may achieved by extending the IGP to advertise the FN authentication capabilities. The most straightforward way to achieve this is to extend the Router Capability TLVs available both in OSPF [RFC4970] and in IS-IS [RFC4971].

4.2.1.1. Area-scoped and Link-scoped Authentication

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

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

4.2.1.2. Simple Password Authentication

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

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

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

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

4.2.1.3. Cryptographic Authentication for FN

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

4.2.1.3.1. MD5

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

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   AuLength    |    Key ID     |             Unused            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Message Digest (bytes 1-4)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Message Digest (bytes 5-8)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Message Digest (bytes 9-12)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Message Digest (bytes 13-16)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

AuLength
   AuLength is set to 20 bytes.

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

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

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

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

4.2.1.3.2. SHA256

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   AuLength    |    Key ID     |             Unused            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Message Digest (bytes 1-4)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Message Digest (bytes 5-8)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              . . .                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Message Digest (bytes 25-28)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Message Digest (bytes 29-32)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Authentication field in FN packets with MD5 cryptographic authentication.

AuLength
   AuLength is set to 36 bytes.

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

Message Digest
   The 32 bytes long SHA256 value calculated on an object which is the concatenation of the FN message, including the FN header, and the pre-shared secret key identified by Key ID.

When receiving an FN message, if the FN header indicates SHA256 authentication, then the last 68 bytes of the FN message are set aside. The recipient forwarding plane element calculates a new SHA256 digest of the remainder of the FN message to which it appends its own known secret key identified by Key ID. The calculated and received digests are compared. In case of mismatch, the FN message is discarded.

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

4.2.1.3.3. Digital Signatures

A router may choose to use public key cryptography to digitally sign the notification to provide certification of authenticity. This mechanism can avoid shared secret that is required for other authentication mechanisms described in this document. This authentication mechanism resembles the authentication mechanism of OSPF with digital signatures as defined in [RFC2154].

5. Security Considerations

This draft has described basic optional procedures for authentication. The mechanism, however, does not protect against replay attacks.

If an application of FN require protection against replay attacks, then these applications should provide their own specific sequence numbering within the FN payload. Recipient applications should accept FN messages only if the included sequence number is valid.

Since the message digest of cryptographic authentication also covers the payload, even if an attacker knew how to construct the new sequence number, it would not be able to generate a correct message digest without the pre shared key. This way, a sequence number in the payload combined with FN’s cryptographic authentication offers sufficient protection against replay attacks.

6. FN Packet Processing Summary

When receiving an FN packet, a node has to perform the following steps.

It has to identify that the packet is an FN packet. This can be done utilising the destination IP address (MC-FN) or by inspecting the UDP port field.

If the flooding like transport logic described in Section 3 is used the node has to perform duplicate check following the teachings in Section 3.1.1.

If AuType is non-null, the node has to perform authentication check as discussed in Section 4.2.1.

To protect against replay attacks, the node shall perform verification of the sequence number provided by the application.

Punt and forward. The notification may need to be multicasted but it also needs to be punted to the local application on the linecard to start processing.

Authentication check, sequence number check and punting/forwarding may commence in any order deemed necessary by the operator. If the operator prefers highest level of security, then both checks should be performed before forwarding. If, however, the operator prefers per-hop performance but still wants to ensure that malice packets cannot harm the network, then authentication and sequence number checks may also happen after punting the packet, i.e. before processing the information contained inside the FN payload. In this case, malicious packets may get propagated to every node but they still do not cause any change in the configuration.

7. IANA Considerations

A UDP port value needs to be assigned by IANA for FN. IANA also needs to maintain values for FN App Type as applications are being proposed.

Multicast addresses used for the distribution trees are either allocated by IANA or they can be a configuration parameter within the local domain.

8. Acknowledgements

The authors owe thanks to Acee Lindem, Joel Halpern and Jakob Heitz for their review and comments. Also thanks to Alia Atlas for constructive feedback.

9. References

9.1. Normative References

[I-D.enyedi-rtgwg-mrt-frr-algorithm]
Envedi, G., Csaszar, A., Atlas, A., cbowers@juniper.net, c., and A. Gopalan, "Algorithms for computing Maximally Redundant Trees for IP/LDP Fast- Reroute", draft-enyedi-rtgwg-mrt-frr-algorithm-03 (work in progress), July 2013.

[I-D.ietf-rtgwg-mrt-frr-architecture]
Atlas, A., Kebler, R., Envedi, G., Csaszar, A., Tantsura, J., Konstantynowicz, M., and R. White, "An Architecture for IP/LDP Fast-Reroute Using Maximally Redundant Trees", draft-ietf-rtgwg-mrt-frr-architecture-03 (work in progress), July 2013.

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

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

[RFC4970] Lindem, A., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 4970, July 2007.

[RFC4971] Vasseur, JP., Shen, N., and R. Aggarwal, "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, July 2007.

[RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, October 2007.

9.2. Informative References

[Eny2009] Enyedi, G., Retvari, G., and A. Csaszar, "On Finding Maximally Redundant Trees in Strictly Linear Time, IEEE Symposium on Computers and Communications (ISCC)", 2009.

[I-D.csaszar-ipfrr-fn]
Csaszar, A., Envedi, G., Tantsura, J., Kini, S., Sucec, J., and S. Das, "IP Fast Re-Route with Fast Notification", draft-csaszar-ipfrr-fn-03 (work in progress), June 2012.

[RFC2154] Murphy, S., Badger, M., and B. Wellington, "OSPF with Digital Signatures", RFC 2154, June 1997.

Appendix A. Further Options for Transport Logic

The options described in this appendix represent alternative solutions to the flooding based approach described in Section Section 3.

It is left for WG discussion and further evaluation to decide whether any of these options should potentially be preferred instead of redundant trees.

A.1. Multicast Tree-based Transport

One way of transporting an identical piece of information to several receivers at the same time is to use multicast distribution trees. A tree based transport solution is beneficial since multicast support is already implemented in all forwarding entities, so it is possible to use existing implementations.

With multicast or tree based transport, the Fast Notification (FN) packet can be recognized by a pre-configured or well known destination IP address, denoted by MC-FN in the following, which is the group address of the FN service.

If the FN service is triggered to send out a notification, the notification will be encapsulated in a new IP packet, where the destination IP address is set to MC-FN.

A.1.1. Fault Tolerance of a Single Distribution Tree

Several solutions described in this draft use a single tree to disseminate a notification from one given source.

The single tree solution is simple, however it is not redundant: a single failure may partition the tree, which will prevent notifications from reaching some nodes in the area.

Different applications may have different needs for reliability. For example, when we use fast notification to disseminate network failure information, all nodes surrounding the failure can detect and originate the failure notifications independently. Any one of these notifications (or a subset of them) may be sufficient for the application to make the right decision. This draft provides several different transport options from which an applications can choose.

A.1.2. Pair of Redundant Trees

If an FN application needs the exact same data to be distributed in the case of any single node or any single link failure, the FN service could opt to run in "redundant tree mode".

A pair of "maximally redundant trees" [I-D.enyedi-rtgwg-mrt-frr-algorithm] ensures that at each single node or link failure each node still reaches the common root of the trees through at least one of the trees. A redundant tree pair is a known prior-art graph-theoretical object that is possible to find on any 2-node connected network. Even better, it is even possible to find maximally redundant trees in networks where the 2-node connected criterion does not "fully" hold (e.g. there are a few cut vertices) [Eny2009], [I-D.ietf-rtgwg-mrt-frr-architecture].

Note that the referenced algorithm(s) build a pair of trees considering a specific root. The root can be selected in different ways, the only thing that is important that each node makes the same selection, consistently. For instance, the node with the highest or lowest router ID can be used.

```
         #1 tree                              #2 tree
         +---+       +---+                    +---+       +---+
         | B |=======|   |                    | B |=======|   |
         +---+       +---+                    +---+       +---+
         //               \\                 //                 \
         //                 \\               //                   \
    +---+                     +---+      +---+                     +---+
    | A |---------------------| R |      | A |=====================| R |
    +---+                     +---+      +---+                     +---+
         \                  //               \\                   /
         \                //                 \\                 /
         +---+       +---+                    +---+       +---+
         |   |=======|   |                    |   |=======|   |
         +---+       +---+                    +---+       +---+
```

Figure 6: Example: a pair of redundant trees (double lines) of a common root R

There is one special constraint in building the redundant trees. A (maximally) redundant tree pair is needed, where in one of the trees the root has only one child in order to protect against the failure of the root itself. Algorithms presented in [Eny2009], [I-D.enyedi-rtgwg-mrt-frr-algorithm] produce such trees.

In redundant-tree mode, each node multicasts the requested notification on both trees, if it is possible, but at least along one of the trees. Redundant trees require two multicast group addresses. MC-FN identifies one of the trees, and MC-FN-2 identifies the other tree.

Each node multicast forwards the received notification packet (on the same tree). The root node performs as every other node but in addition it also multicast the notification on the other tree! I.e. it forwards a replica of the incoming notification in which it replaces the destination address identifying the other multicast distribution tree.

When the network remains connected and the root remains operable after a single failure, the root will be reached on at least one of the trees. Thus, since the root can reach every node along at least one of the trees, all the notifications will reach each node. However, when the root or the link to the root fails, that tree, in which the root has only one child, remains connected (the root is a leaf there), thus, all the nodes can be reached along that tree.

For example, let us consider that in Figure 6 FN is used to disseminate failure information. If link A-B fails, the notifications originating from node B (e.g. reporting that the connectivity from B to A is lost) will reach R on tree #1. Notifications originating from A (e.g. reporting that the connectivity from A to B is lost) will reach R on tree #2. From R, each node is reachable through one of the trees, so each node will be notified about both events.

A.2. Unicast

This method addresses the need in a unique way. It has the following properties:

Plain simple, without the need of any forwarding plane change or cooperation;

Short turnaround time (i.e. ready for next hit);

100% link break coverage (may not work in certain node failure cases);

Little change to OSPF (need encapsulation for IS-IS).

A.2.1. Method

The method is simple in design, easy to implement and quick to deploy. It requires no topology changes or specific configurations. It adds little overhead to the overall system.

The method sends the event message to every router in the area in an IP packet. This appears burdensome to the sending router which has to duplicate the packet sending effort many times. Practical experience has shown, however, that the amount of effort is not a big concern in reasonable sized networks.

Normal flooding (regular or fast) process requires a router to duplicate the packet to all flooding eligible interfaces. All routers have to be fast-flooding-aware. This implies new code to every router in control plane and/or forwarding plane.

The method uses a different approach. It takes advantage of the given routing/forwarding table in each router in the IP domain. The originating router of the flooding information simply sends multiple copies of the packet to each and every router in the domain. These packets are forwarded to the destination routers at forwarding plane speed,

just like the way the regular IP data traffic is handled. No special handling in any other routers is needed.

This small delay on the sender can be minimized by pre-downloading the link-broken message packets to the forwarding plane. Since the forwarding plane already has the list of all routers which are part of the IGP routing table, the forwarding plane can dispatch the packet directly.

In essence, the flooding in this method is tree based, just like a multicast tree. The key is that no special tree is generated for this purpose; the normal routing table which is an SPF tree (SPT) plays a role of the flooding tree. This logic guarantees that the flooding follows the shortest path and no flooding loop is created.

A.2.2. Sample Operation

Figure 7 depicts a scenario where router A wants to flood its message to all other routers in the domain using the unicast flooding method.

Instead of sending one packet to each of its neighbor, and letting the neighbor flood the packet further, router A directly send the same packet to each router in the domain, one at a time. In this sample network, router A sends out 5 packets.

```
A---B---C---D
\
--E---F

1. Packet(A->B);
2. Packet(A->C);
3. Packet(A->D);
4. Packet(A->E);
5. Packet(A->F).
```

Figure 7: Multiple Unicast Packets

The unicast flooding procedure is solely controlled by the sending router. No action is needed from other routers other than their normal forwarding functionalities. This method is extremely simple and useful for quick prototyping and deployment.

A.3. Gated Multicast through RPF Check

This method fulfills the purpose with the following characters:

1. No need to build the multicast tree. It is the same as the SPT computed by the IGP routing process;

2. Flooding loops are prevented by RPF Check.

The method has all the benefits of multicast flooding. It, however, does not require running multicast protocol to setup the multicast tree. The unicast shortest path tree is used as a multicast tree.

A.3.1. Loop Prevention - RPF Check

In this mechanism, the distribution tree is not explicitly built. Rather, each node will first do a Reverse Path Forwarding (RPF) check before it floods the notification to other links.

A special multicast address is defined and is subject to IANA approval. This address is used to qualify the notification packet for fast flooding. When a notification packet arrives, the receiving node will perform an IP unicast routing table lookup for the originator IP address of the notification and find the outgoing interface. Only when the arriving interface of the notification is the same as the outgoing interface leading towards the originator IP address, will the notification be flooded to other interfaces.

IP Multicast forwarding with RPF check is available on most of the routing/switching platforms. To support flooding with RPF check, a special IP multicast group must be used. A bi-directional IP multicast forwarding entry is created that consists of all interfaces within the flooding scope, typically an IGP area.

A.3.2. Operation

The Gated flooding operation is illustrated in Figure 8.

```
           All Routers, IGP Process:
           if (SPT ready) {
            duplicate the SPT as Bidir_Multicast_tree;
            download the multicast_tree to forwarding plane;
           }
           add FNF_multicast_group_addr;

               Sender of the FNF notification:
           if (breakage detected) {
            pack the notification in a packet;
            send the packet to the FNF_multicast_group_addr;
           }

           Receiver of the FNF notification:
            if (notification received) {
            if (RPC_interface == incoming_interface) {
             multicast the notification to all other interfaces;
            }
            forward the notification to IGP for processing;
```

```
              }
```

Figure 8: Gated flooding operation

Figure 9 shows a sample operation on a four-router mesh network. The left figure is the topology. The right figure is the shortest path tree rooted at A.

Router A initiates the flooding. But the downstream routers B, C, and D will drop all messages except the ones that come from their shortest path parent node. For example, A’s message to C via B is dropped by C, because C knows that its reverse path forwarding (RPF) nexthop is A.

```
                A       A
               /|\     / \
               B---C   B   C
               \|/     \
                D       D
```

Figure 9: Loop Prevention through the RPF check

A.4. Further Multicast Tree based Transport Options

A.4.1. Source Specific Trees

One implementation option is to rely on source specific multicast. This means that even though there is only a single multicast group address (MC-FN) allocated to the FN service, the FIB of each router is configured with forwarding information for as many trees as many FN sources (nodes) there are in the routing area, i.e. to each (S_i ,MC-FN) pair.

A.4.2. A Single Bidirectional Shared Tree

In the previous solution each source specific tree is a spanning tree. It is possible to reduce the complexity of managing and configuring n spanning trees in the area by using bidirectional shared trees. By building a bidirectional shared tree, all nodes on the tree can send and receive traffic using that single tree. Each sent packet from any source is multicasted on the tree to all other receivers.

The tree must be consistently computed at all routers. For this, the following rules may be given:

The tree can be computed as a shortest path tree rooted at e.g. the highest router-id. When multiple paths are available, the neighbouring node in the graph e.g. with highest router-id can be picked. When multiple paths are available through multiple interfaces to a neighbouring node, e.g. a numbered interface may be preferred over an unnumbered interface. A higher IP address may be preferred among numbered interfaces and a higher ifIndex may be preferred among unnumbered interfaces.

Note, however, that the important point is that the rules are consistent among nodes. That is, a router may pick the lower router IDs if it is ensured that ALL routers will do the same to ensure consistency.

Multicast forwarding state is installed using such a tree as a bi-directional tree. Each router on the tree can send packets to all other routers on that tree.

Note that the multicast spanning tree can be built using [RFC5015] so that each router within an area subscribes to the same multicast group address. Using BIDIR-PIM in such a way will eventually build a multicast spanning tree among all routers within the area. (BIDIR-PIM is normally used to build a shared, bidirectional multicast tree among multiple sources and receivers.)

A.5. Layer 2 Networks

Layer 2 (e.g. Ethernet) networks offer further options for distributing the notification (e.g. using spanning trees offered by STP). Definition of these is being considered and will be included in a future revision of this draft.

Authors’ Addresses

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

Email: Wenhu.Lu@ericsson.com

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

Email: Sriganesh.Kini@ericsson.com

Andras Csaszar (editor)
Ericsson
Irinyi J utca 4-10
Budapest 1117
Hungary

Email: Andras.Csaszar@ericsson.com

Gabor Sandor Enyedi
Ericsson
Irinyi J utca 4-10
Budapest 1117
Hungary

Email: Gabor.Sandor.Enyedi@ericsson.com

Jeff Tantsura
Ericsson
300 Holger Way
San Jose, California 95134
USA

Email: Jeff.Tantsura@ericsson.com