

Routing Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: July 28, 2014

IJ. Wijnands, Ed.  
L. De Ghein  
Cisco  
G. Enyedi, Ed.  
A. Csaszar  
J. Tantsura  
Ericsson  
January 24, 2014

Tree Notification to Improve Multicast Fast Reroute  
draft-wijnands-rtgwg-mcast-frr-tn-02

## Abstract

This draft proposes dataplane triggered Tree Notifications to support multicast fast reroute for PIM and mLDP. These Tree Notifications are initiated by a node detecting the failure to a Repair Node downstream. A Repair Node is a node that has a pre-built backup path that can circumvent the failure. Using this mechanism, a Repair Node has the ability to learn about non-local failures quickly without having to wait for the IGP to convergence. This draft also covers an optional method to avoid bandwidth usage on the pre-built backup path.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on July 28, 2014.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal

## Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Terminology and Definitions</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Introduction</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Improving Non-local failures</a>	<a href="#">4</a>
<a href="#">3.1.</a>	<a href="#">Downstream Tree Notifications</a>	<a href="#">5</a>
<a href="#">3.2.</a>	<a href="#">DTN processing logic</a>	<a href="#">5</a>
<a href="#">3.3.</a>	<a href="#">Repair Node discovery</a>	<a href="#">7</a>
<a href="#">3.3.1.</a>	<a href="#">Repair Node Information item</a>	<a href="#">8</a>
<a href="#">4.</a>	<a href="#">Reduce the bandwidth consumption in networks with fast failover response times</a>	<a href="#">8</a>
<a href="#">4.1.</a>	<a href="#">Joining a secondary tree in blocking mode</a>	<a href="#">9</a>
<a href="#">4.2.</a>	<a href="#">Upstream Tree Notifications</a>	<a href="#">9</a>
<a href="#">5.</a>	<a href="#">MRT/MCI-Only Mode</a>	<a href="#">10</a>
<a href="#">6.</a>	<a href="#">TN Authentication</a>	<a href="#">10</a>
<a href="#">7.</a>	<a href="#">The TN Packet</a>	<a href="#">11</a>
<a href="#">7.1.</a>	<a href="#">TN Packet Format</a>	<a href="#">11</a>
<a href="#">7.1.1.</a>	<a href="#">TN TimeStamp TLV Format</a>	<a href="#">13</a>
<a href="#">7.1.2.</a>	<a href="#">TN Signature TLV Format</a>	<a href="#">13</a>
<a href="#">8.</a>	<a href="#">PIM Specific TN Components</a>	<a href="#">14</a>
<a href="#">8.1.</a>	<a href="#">RNI item in PIM Join Message</a>	<a href="#">14</a>
<a href="#">8.2.</a>	<a href="#">Tree Information Item</a>	<a href="#">16</a>
<a href="#">8.3.</a>	<a href="#">Incremental deployment</a>	<a href="#">17</a>
<a href="#">9.</a>	<a href="#">mLDP Specific TN Components</a>	<a href="#">17</a>
<a href="#">9.1.</a>	<a href="#">RNI item in mLDP Label Mapping</a>	<a href="#">18</a>
<a href="#">9.2.</a>	<a href="#">Tree Information Item</a>	<a href="#">19</a>
<a href="#">10.</a>	<a href="#">Acknowledgements</a>	<a href="#">19</a>
<a href="#">11.</a>	<a href="#">IANA Considerations</a>	<a href="#">20</a>
<a href="#">12.</a>	<a href="#">Security Considerations</a>	<a href="#">20</a>
<a href="#">13.</a>	<a href="#">References</a>	<a href="#">20</a>
<a href="#">13.1.</a>	<a href="#">Normative References</a>	<a href="#">20</a>
<a href="#">13.2.</a>	<a href="#">Informative References</a>	<a href="#">21</a>
	<a href="#">Authors' Addresses</a>	<a href="#">21</a>

## [1.](#) Terminology and Definitions

MoFRR : Multicast only Fast Re-Route.

LFA : Loop Free Alternate.

mLDP : Multi-point Label Distribution Protocol.

PIM : Protocol Independent Multicast.

UMH : Upstream Multicast Hop, a candidate next-hop that can be used to reach the root of the tree.

tree : Either a PIM (S,G)/(\*,G) tree or a mLDP P2MP or MP2MP LSP.

OIF : Outgoing InterFace, an interface used to forward multicast packets down the tree towards the receivers. Either a PIM (S,G)/(\*,G) tree or a mLDP P2MP or MP2MP LSP.

IIF : Incoming InterFace, an interface where multicast traffic is received by a router.

MCE : MultiCast Egress, the last node where the multicast stream exits the current transport technology (MPLS-mLDP or IP-PIM) domain or administrative domain. This maybe the router attached to a multicast receiver.

MCI : MultiCast Ingress, the node where the multicast stream enters the current transport technology (MPLS-mLDP or IP-PIM) domain. This maybe the router attached to the multicast source.

DTN : Downstream Tree Notification.

UTN : Upstream Tree Notification.

TN : Tree Notification, Upstream or Downstream

JM : Join Message, the message used to join to a multicast tree, i.e. to build up the tree. In PIM, this is a JOIN message, while in mLDP this corresponds to a Label Mapping message.

MRT : Maximally Redundant Trees.

Repair Node : The node performing a dual-join to the tree through two different UMHS. Sometimes also called as dual-joining node or merging node (it merges the secondary and primary tree).

RNI : The Repair Node Information is an item included in the TN which holds the necessary repair information when the TN is sent to the Repair Node.

Branching Node : A node, (i) which is considered as being on the primary tree by its immediate UMH and (ii) which has at least one OIF on the secondary tree installed for a multicast tree.

## 2. Introduction

Both [[I-D.karan-mofrr](#)] and [[I-D.atlas-rtgwg-mrt-mc-arch](#)] describe "live-live" multicast protection, where a node joins a tree via different candidate upstream multicast hops (UMH). With MoFRR the list of candidate UMHS can come from either ECMP or Loop Free Alternate (LFA) paths towards the MultiCast Ingress node (MCI). With MRT, the candidate UMHS are determined by looking up the MCI in two different (Red and Blue) topologies. In either case, the multicast traffic is simultaneously received over different paths/topologies for the same tree. The node 'dual-joining' the tree needs a mechanism to prevent duplicate packets being forwarded to the end user. For that reason a node 'dual-joining' the tree only accepts packets from one of the UMHS at the time. Which UMH is preferred is a local decision that can be based on IGP reachability, link status, BFD, traffic flow monitoring, etc...

Should the node detect a local failure on the primary UMH, the node has an instantly available secondary UMH that it can switch to, simply by unblocking the secondary UMH. The dual-joining node is

also called Repair Node in the following.

This draft attempts to improve these solutions by:

- o Improving fail-over time and the reliability of failure detection for non-local failures; and
- o Reducing the bandwidth consumption in a network with fast failover response times, by avoiding sending the multicast traffic over the secondary path.

### [3.](#) Improving Non-local failures

If a failure is not local and happens further upstream, the dual-joining node needs a fast mechanism (i) to detect the upstream failure and (ii) to learn that other upstream nodes cannot circumvent the failure. Existing methods based on traffic monitoring are limited in scope and work best with a steady state packet flow.

Therefore, we propose a method which can trigger the unblocking the secondary UMH independently of the packet flow.

Figure 1 shows an example. Consider that, e.g., node A goes down. Nodes C, D and E cannot detect that locally, so they need to resort to other means. After detecting the failure, node C should not change to its secondary UMH (node J) as it won't help for the failure of A. Node D, on the other hand, will have to unblock its secondary UMH (node I). Yet again, with MoFRR, node E should not unblock its secondary UMH (node K): (i) this won't help in resolving the failure of node A, and (ii) one of its upstream nodes (node D in this case) will be able to restore the stream with a fail-over action.

#### [3.1.](#) Downstream Tree Notifications

When a node detects a local failure of its primary UMH it MUST originate a Downstream Tree Notification (DTN) to all the Repair Nodes directly below it in the multicast tree. The method of discovering such nodes is described in [Section 3.3](#). When a Repair Node receives a DTN containing the primary UMH of the node, it must switch to the secondary UMH.

DTN packets are sent to the Repair Node via unicast. The packet may be forwarded using any transport that is available (MPLS or IP) to reach the destination. The IP precedence in the IP header should have a value of 6 (Internetwork Control). The EXP field (Traffic Class field) in the MPLS header should have a value of 6. The DTN packets are identified by a well known port number (to be allocated). Using a well-known port number it is easy for the Repair Node to identify the DTN packet and invoke the procedures as described in this draft. We are proposing to allocate different port numbers for PIM and mDLP since it will be easier to dispatch the packet to the right process dealing with this request.

When a router detects a local failure, it should sent out the DTN packet to the Repair Node as fast as possible. The sooner the Repair Node gets the packet, the sooner the traffic can be restored. It is recommended that the DTN packet is pre-created and originated from the data-plane. The same is true for receiving the DTN packet on the Repair Node, the faster it can be processed, the faster the traffic is restored. For both forwarding and processing the DTN, control-plane interaction SHOULD be avoided to get the best failover results.

### [3.2.](#) DTN processing logic

When a DTN packet is received on the Repair Node it must determine which tree and UMH the notification is for. The information encoded in the DTN is specific for the type of tree being used, i.e. PIM vs

mLDP. For details on the specific encoding see [Section 8](#) and [Section 9](#) for the details. Once the Repair node has determined the tree and the UMH, the following rules are use for processing the DTN.

1. If the UMH encoded in the DTN packet is the primary UMH in the tree, the secondary UMH MUST become the new primary UMH and the old primary MUST become the secondary.
2. If the UMH encoded in the DTN packet is the secondary UMH in the tree, no action needs to be taken.
3. If a DTN notification has been received for both the primary and secondary UMH in the tree, a new DTN notification MUST be originated to the Repair Node(s) downstream from this node.

In order for the Repair Node to determine that a DTN notification was received for both the primary and secondary UMH, it must store the fact a DTN was received for a particular UMH.

Consider the example in Figure 1 below. MCI is the root of a tree that includes the nodes as follows (based on the primary UMH).

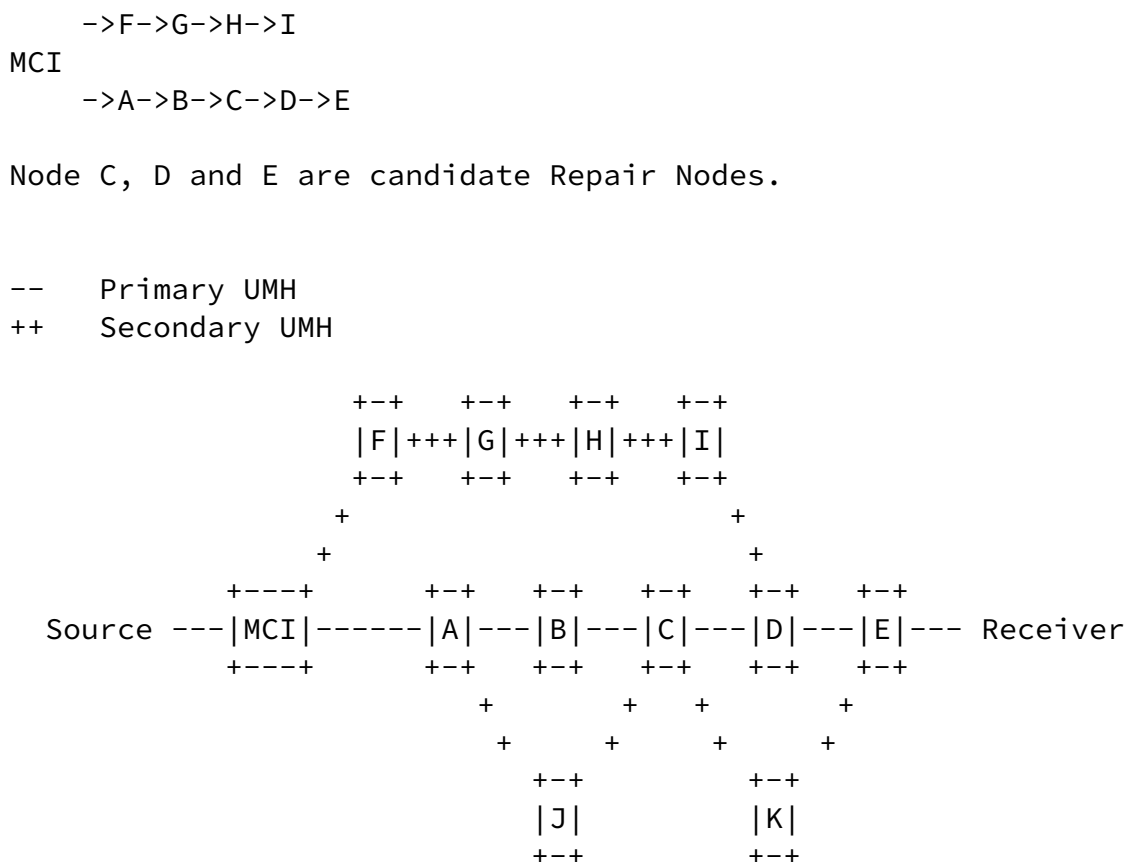


Figure 1: Remote failure example

Suppose that the link between node A and B failed, B is directly connected and will detect the failure locally. In this case, node B is the only node that detects the failure and will originate a DTN to its downstream repair node C. Node C will receive the DTN for the UMH that is the primary UMH. Following rule 1 ([Section 3.2](#)), node C will make the backup UMH the new primary. No further action is needed because C has repaired the tree via node J. Note, J would not have sent a DTN to node C because J is not directly connected to the

failing link.

Suppose that node A fails, B and J are directly connected and detect the failure locally. A DTN packet is triggered to first downstream repair node of A, which is node C. Node C is an unusable Repair Node because it will receive DTN for both the primary UMH (from B) and the secondary UMH (from J). Following rule 3 ([Section 3.2](#)), C can't repair the tree and must sent a new DTN packet towards the Repair Nodes of C, which are D, on the primary path, and E, on the secondary path.

Suppose that the link between A and the MCI failed. Node A is directly connected to the failure and will trigger a DTN packet to its downstream repair node(s). In this case, node A has learned about the downstream repair node C twice, the primary UMH (via node B) and secondary UMH (via node J). Node A will therefore sent a DTN packet including both the primary and secondary UMH to node C (see [Section 7](#) for details on the encoding). Following rule 3 ([Section 3.2](#)), C can't repair the tree and must sent a new DTN packet towards the Repair Nodes of C, which are D, on the primary path, and E, on the secondary path.

The DTN packet that D received from C will match against the primary UMH. Following rule 1, D will activate the backup path to I. The DTN packet that E received from C will match against the backup UMH, following rule 2, no action is taken. In the example one can see that we recovered from the failure because node D started accepting the data packets from node I and is forwarding them to node E.

### [3.3.](#) Repair Node discovery

In example Figure 1 we wrote that nodes C, D and E are the repair nodes. How does a node determine that it is a Repair Node? The rule is straightforward, a node that is enabled to join two UMH's, one in active the other in backup ([\[I-D.karan-mofrr\]](#)), is a repair node on the tree. A Repair node has the ability to repair the tree for the nodes upstream from this node. In order for the Repair Node to get notified of upstream failures (ie DTN), the nodes upstream from the Repair Node need to learn about it.

#### [3.3.1.](#) Repair Node Information item

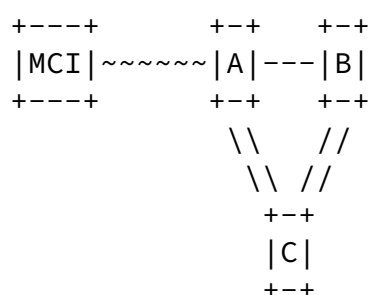


A Repair Node MUST advertise its own address (either a router ID or any directly connected address) and an UMH identifier to the nodes upstream on the tree. This address and UMH are part of the RNI (Repair Node Information) item that is included in the JM. The RNI is carried hop by hop in the JM upstream. If a node along the path is not a Repair Node, it will save the RNI and forward it further upstream. If the node is a Repair Node, it will save the RNI and include its own RNI in the JM sent further upstream. If a Repair Node changes one of its UMH's, it needs to trigger a new RNI to its upstream node(s) to notify them of the changed UMH. If a RNI is received and it does not match the saved RNI, the new RNI overrides the old RNI and triggers a JM with the new RNI to its upstream node(s). A RNI includes protocol specific information on how to identify the tree and UMH. For that reason it is documented in the protocol specific sections [Section 8](#) and [Section 9](#).

The Repair Node MAY include additional information in the RNI for reasons of security and robustness, please see [Section 6](#) and [Section 7.1](#).

4. Reduce the bandwidth consumption in networks with fast failover response times

In some of networks, such as aggregation networks, bandwidth is more sparse than, e.g., in core networks. Live-live multicast protection results in more bandwidth consumption in the network as it continuously pulls traffic on both trees. In such networks it is relevant if the capacity serving backup purposes can be used, most of the time, by best-effort or even by lower-than-best-effort traffic.



Nodes A and B have receivers. Double lines show bandwidth consumption that is superfluous when there is no failure in the network.

Figure 2: Example for secondary segments occupying bandwidth in MoFRR

In live-standby mode the aim is that the secondary tree is not forwarding multicast traffic as long as there is no failure. In order to achieve such a "live-standby" multicast protection the following procedures must be followed:

- o Upstream nodes block their OIF when they are part of a standby tree.
- o If all of the OIF's of the node are marked as blocking, the node joins the tree in blocking mode further upstream.
- o A procedure so that the upstream node can quickly unblock its OIF and starts to forward.

#### [4.1.](#) Joining a secondary tree in blocking mode

The JM sent to the secondary UMH includes an identifier to indicate the upstream node MUST not forward packets down this branch of the tree. The identifier is TBD. The mechanism to join a secondary path is identical to what the MRT and MoFRR drafts describe, i.e. a Repair Node simply sends a secondary JM through another UMH (on another topology, in case of MRT). If a node receives a JM without a blocking identifier for an OIF that previously was in blocking mode, the blocking mode is reset and the node starts forwarding out of this interface. If this node joined the tree in blocking mode further upstream, a new JM MUST be originated to reset the blocking state further upstream.

#### [4.2.](#) Upstream Tree Notifications

In order to make an upstream node start forwarding on the backup path quickly after a failure was detected on the primary UMH, we sent an Upstream Tree Notification (UTN) to the upstream node on the backup UMH. The failure on the primary UMH may be local or detected using a DTN. The UTN received by the upstream node should be processed in the data-plane and reset the blocking state of the OIF. If this node also joined the tree in blocking mode upstream, a UTN has to be forwarded further upstream. This procedure is repeated until we find a node that is not in blocking mode or we reached the MCI.

When the upstream node resets the blocking mode in the data-plane, the control plane will still have the blocking mode set. In order for the control plane to get in sync with the data-plane, the node that originated the UTN MUST also trigger a JM without blocking mode.

The upstream node receiving the UTN must be able to identify the tree

which the notification is sent for, as well as the downstream interface it applies to. The information is encoded in a same RNI

item that is used for DTN packets. For details please see the protocol specific sections [Section 8](#) and [Section 9](#).

Like DTN packets, UTN packets are sent via unicast to the upstream node.

## [5.](#) MRT/MCI-Only Mode

If each node in the network supports UTN and also all nodes support MRT, the nodes may work in "MRT/MCI-only" mode.

In MRT/MCI-only mode, there is one single Repair Node for all failures, the MCI. Other nodes MUST NOT consider themselves as Repair Nodes. MRT ensures the necessary maximally disjoint secondary tree up to the MCI, on a second topology. Only the MCI MUST keep its OIFs corresponding to the secondary tree blocked. Similarly, only MCEs MUST keep their secondary backup IIFs blocked. Any other nodes MUST NOT block their (secondary) IIFs or OIFs.

In MRT/MCI-only mode, the UTNP MUST be forwarded directly to the MCI. This mode enables that a node detecting a downstream failure of the primary tree MAY send a UTNP upstream towards the source/MCI on the primary tree.

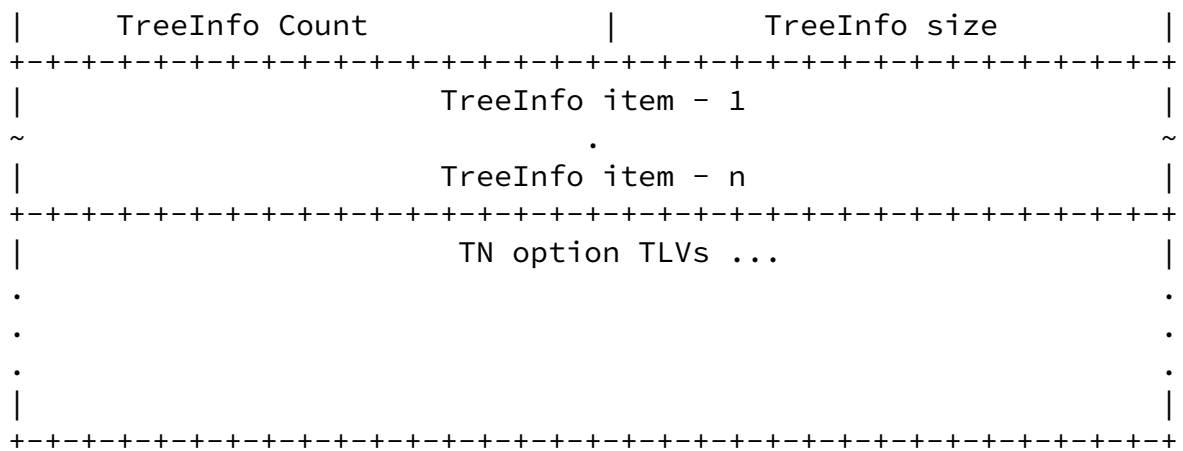
If an UTNP is received by the MCI on the secondary topology in "MRT/MCI-only" mode, the MCI MUST unblock the OIF where the UTNP was received. This activates a whole sub-tree of the secondary tree.

If an UTNP is received by the MCI on the primary topology in "MRT/MCI-only" mode, the MCI gets no information on which leg to activate on the secondary tree, so it MUST activate (unblock) all secondary legs.

## [6.](#) TN Authentication

If a malicious attacker can reproduce the TN packet format, unwanted reconvergence can be triggered. In order to avoid such attack, a TN





Version number: This is a 1 octet field encoding the version number, currently 0.

Address Family: This is a 2 octet field encoding a value from ADDRESS FAMILY NUMBERS in [\[RFC3232\]](#) that encodes the address family for the Root Address of the tree.

Type: This is a 1 octet field encoding the message type, currently two are defined;

Type 0: DownstreamTree Notification.

Type 1: Upstream Tree Notification.

Originator ID: 4 bytes long unique ID of the originator. That can be some loopback IPv4 address if there is such, or can be set by the operator.

Sequence Number: Number unique for each failure case. It is recommend to start at 0, and to be increased by 1 each time a new TN is originated. The Sequence number may differ at each node, thus the sender and the receiver must know the same sequence number.

TreeInfo count: 2 octet field encoding the number of TreeInfo items

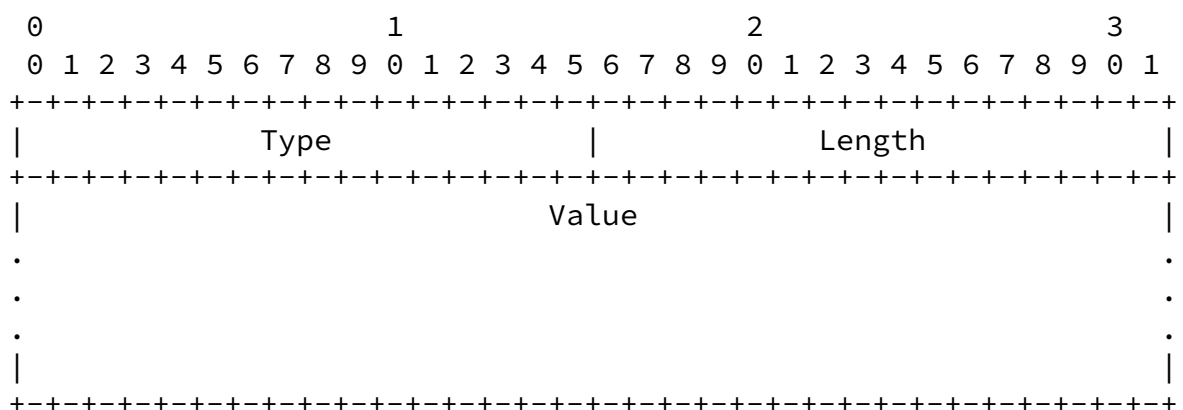
includes.

TreeInfo size: 2 octet field encoding the number of octets use to encode the TreeInfo's following.

TreeInfo item: The encoding of this field is protocol specific, see [Section 8](#) and [Section 9](#).

TN option TLVs: TLVs (Type-Length-Value tuples) describing additional options for TN packets.

The TLV's have the following format.



Type: This is a 2 octet field encoding the type number of the TLV.

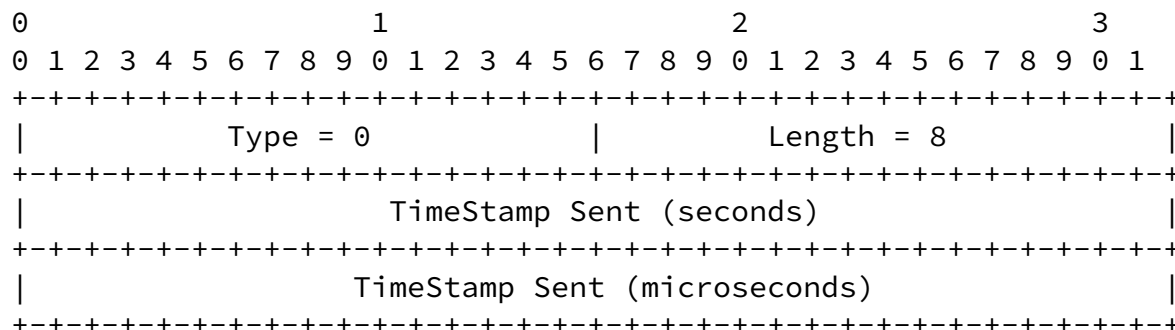
Length: This is a 2 octet field encoding the length of the Value in octets.

Value: String of Length octets, to be interpreted as specified by

the Type field.

#### 7.1.1. TN TimeStamp TLV Format

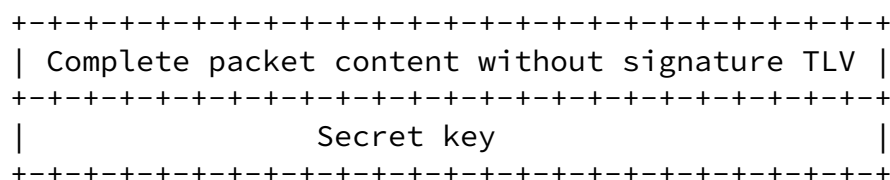
The TimeStamp is an optional TLV that MAY be included when the TN was originated, it has the following format.



TimeStamp: The TimeStamp is the time-of-day (in seconds and microseconds, according to the sender's clock) in NTP format [NTP] when the Tree Notification is sent.

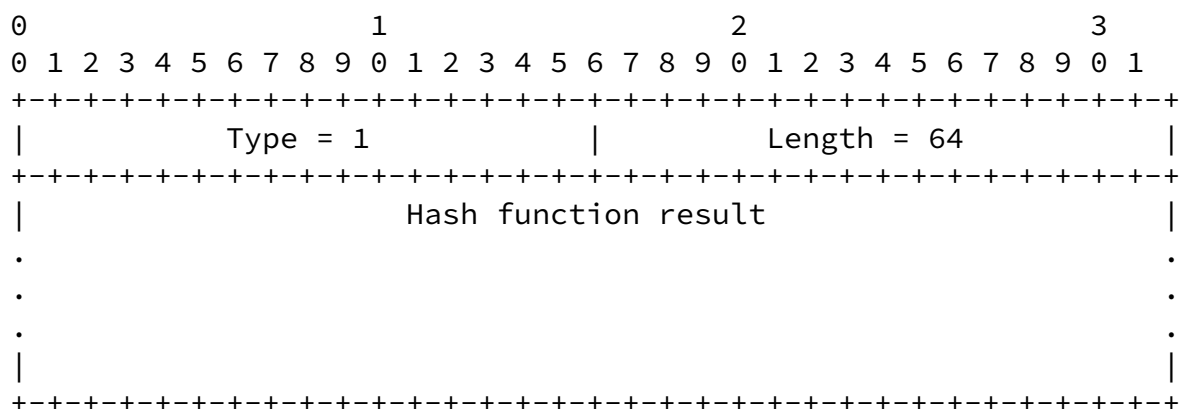
#### 7.1.2. TN Signature TLV Format

TN Signature is an optional TLV, which protects the whole TNP (including other TLVs) against attacks thus it must be the last TLV if present. The signature is SHA-512 hash value. The input of the hash function is as follows:



Signature input: The input of the hash function is the packet extended with TN security key

The build up of the TLV is as follows:



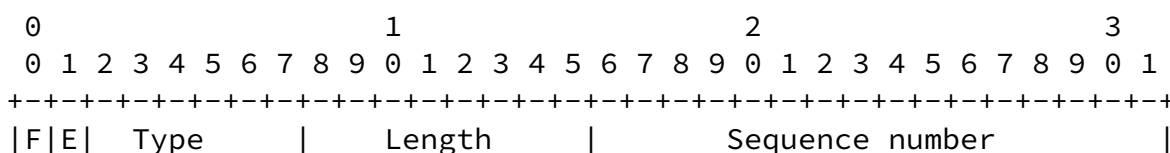
Signature: SHA-512 signature protecting TN packets.

## 8. PIM Specific TN Components

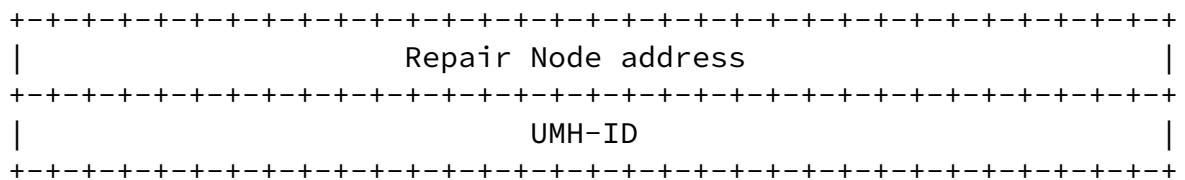
In this section we are documenting the PIM specific data-structures and procedures (if they are different from the generic procedures are defined in this document). As described in this document, TN packets are UDP/IP packets sent via unicast to its destination. The UDP port number for PIM is set to the (to be) assigned IANA port number for PIM-TN.

### 8.1. RNI item in PIM Join Message

As described previously, PIM must insert the RNI when sending a PIM join to its UMH. The RNI includes its router ID, sequence number and UMH Identifier. The UMH-ID can be locally unique identifier since its has only local significance on the Repair Node. A good ID to use would be the IP address of the interface associated with the UMH the PIM join is sent to. The RNI is carried in the PIM Join as a new PIM Attribute following [\[RFC5384\]](#). The PIM RNI attribute has the following format for IPv4.



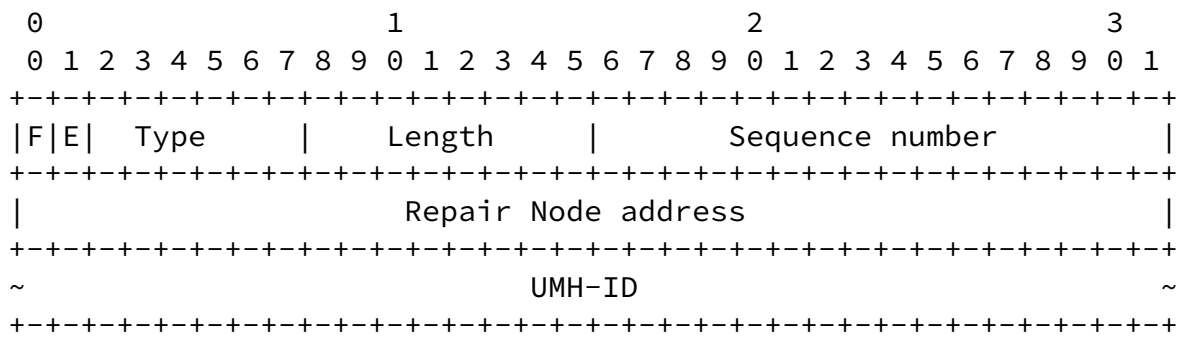




- F Forward if not understood.
- E End of Attributes, following [\[RFC5384\]](#).
- Type: This 6 bit field should be assigned by IANA for TN specific JOIN messages.
- Length: Length = 10 octets.
- Sequence number: 2 octets long field, describing the sequence number of the sending Repair Node.
- Repair Node address: The router ID of the Repair Node, in IPv4 address format.
- UMH-ID: This is a 4 octet field encoding UMH identifier. This is the IPv4 address of the interface associated with the UMH the PIM join is sent to.

Figure 3: PIM IPv4 RNI attribute TLV

The PIM RNI attribute has the following format for IPv6.



F Forward if not understood.

E End of Attributes, following [\[RFC5384\]](#).

Type: This 6 bit field should be assigned by IANA for TN specific JOIN messages.

Length:   Length = 16 octets.

Sequence number: 2 octets long field, describing the sequence number of the sending Repair Node.

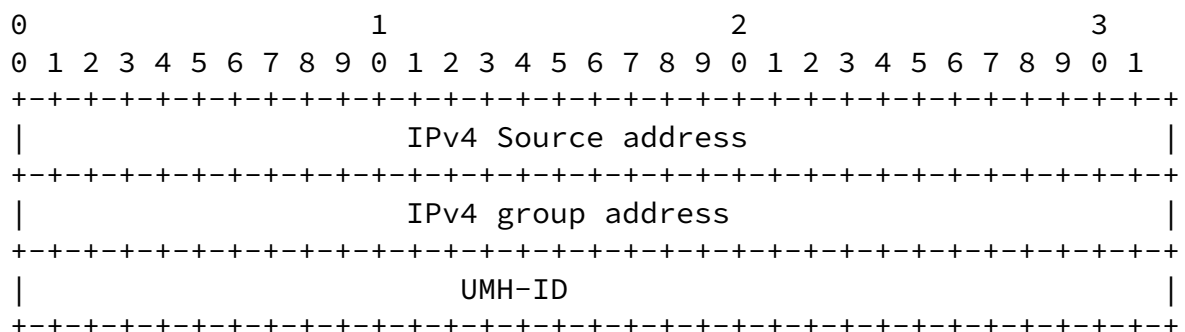
Repair Node address: The router ID of the Repair Node, in IPv4 address format.

UMH-ID: This is a 16 octet field encoding UMH identifier. This is the IPv6 address of the interface associated with the UMH the PIM join is sent to.

Figure 4: PIM IPv6 RNI attribute TLV

## 8.2. Tree Information Item

A TN packet contains one or more TreeInfo items that allows a Merge Node to identify which tree(s) and interface(s) are effected by the TN. The same encoding is used for DTN and UTN packets. The PIM TreeInfo items are defined for IPv4 and IPv6. Which version is to be included in the TN packet depends on Address Family in the TN packet. The UMH-ID included in the DTN MUST be taken from the RNI that was signalled for that tree. The UMH-ID for UTN packets is the PIM neighbor address for that tree. The TreeInfo item has the following format:



Source Address: This is a 4 octet field encoding the IPv4 source address of the tree. A source address of 0.0.0.0 means that this TN relates to a (\*,G) tree.

Internet-Draft

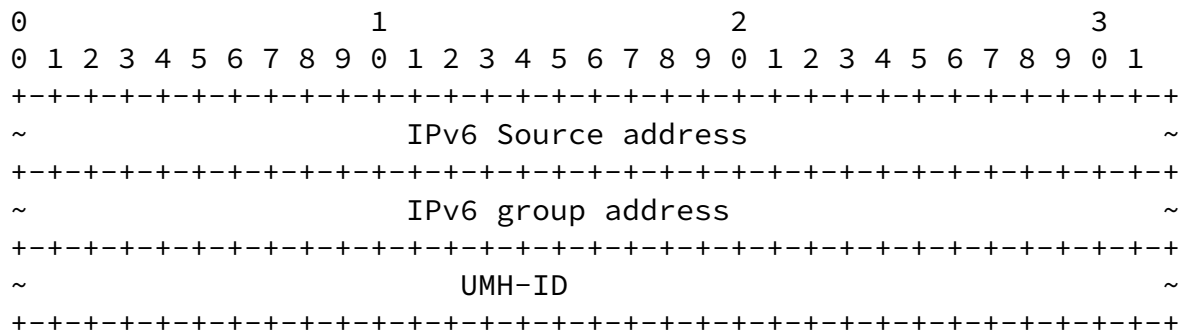
Tree Notification for Multicast FRR

January 2014

Group Address: This is a 4 octet field encoding the IPv4 group address of the tree.

UMH-ID: This is a 4 octet field encoding UMH identifier.

Figure 5: PIM IPv4 TreeInfo item



Source Address: This is a 16 octet field encoding the IPv6 source address of the tree. A source address of 0:0:0:0:0:0:0:0 means that this TN relates to a (\*,G) tree.

Group Address: This is a 16 octet field encoding the IPv6 group address of the tree.

UMH-ID: This is a 16 octet field encoding UMH identifier.

Figure 6: PIM IPv6 TreeInfo item

### [8.3.](#) Incremental deployment

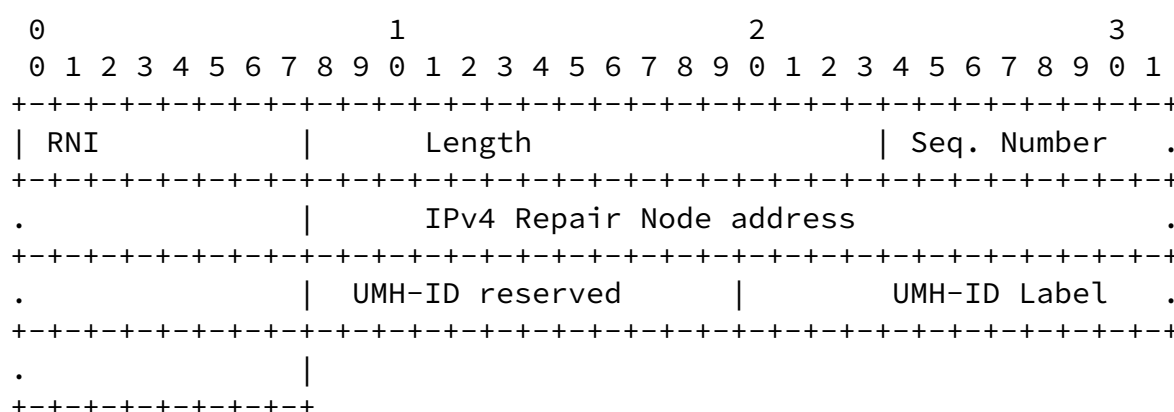
Joins with a RNI can be forwarded through legacy nodes if the Transitive Attribute (see [[RFC5384](#)]) has the F bit set to 1. It is up to the network operator to determine this. The DTN functionality can be deployed incrementally as long as the node detecting the failure and Repair Nodes support it.

## [9.](#) mLDP Specific TN Components

In this section we are documenting the mLDP specific data-structures and procedures (if they are different from the generic procedures are defined in this document). As described in this document, TN packets are UDP/IP packets sent via unicast to its destination. The UDP port number for mLDP is set to the (to be) assigned IANA port number for mLDP-TN.

### 9.1. RNI item in mLDP Label Mapping

The RNI item for mLDP is encoded in a LDP MP Status TLV as documented in [\[RFC6388\] section 5](#). A new LDP MP Status Value Element is created for this purpose and called the RNI Status. The RNI Status includes the router ID, sequence number and UMH Identifier. The UMH-ID can be locally unique identifier since its has only local significance on the Repair node. For mLDP the value that MUST be used is the Local Label associated with the UMH the mLDP Label Mapping is sent to. The RNI status is carried in Label Mapping messages and has the following format.



**RNI Type:** This 1 octet field assigned by IANA for RNI Status Value Element Types.

**Length:** This is a 2 octet field, describing the length of the Value, Length = 10 octets.

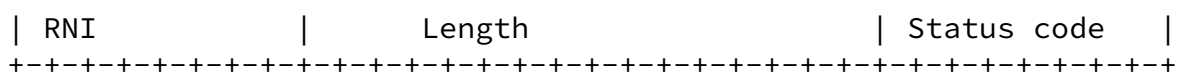
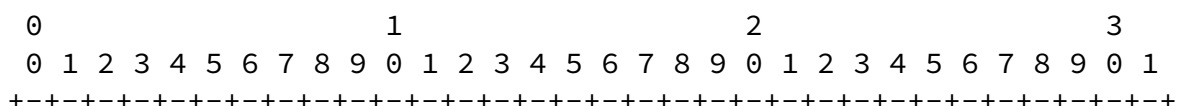
**Sequence number:** 2 octets long field, describing the sequence number of the sending Repair Node.

IPv4 Repair Node address: The IPv4 address of the Repair Node.

UMH-ID reserved: 12 bit field, reserved.

UMH-ID Label: This is a 20 bit field encoding a Label as UMH identifier.

Figure 7: mLDP RNI Status Value Element



MBB Type: Type 1 (to be assigned by IANA)

Length: 1

Status code: 1 = MBB request

2 = MBB ack

## 9.2. Tree Information Item

A TN packet contains one or more TreeInfo items that allows a Merge Node to identify which tree(s) and interface(s) are effected by the TN. The same encoding is used for DTN and UTN packets. Following [\[RFC6388\]](#), mLDP will assign a unique Label to each upstream node per MP-LSP. This label identifies the UMH AND the LSP. Since we are using a label to identify the UMH and LSP, there is no need to define a IPv4 and IPv6 specific encoding. The Label included in the DTN MUST be taken from the RNI that was signalled for that tree. The Label for UTN packets is the Local Label that was allocated for that tree. The TreeInfo item has the following format:



## [12.](#) Security Considerations

Two types of security problems can be foreseen by the authors:

- o Handling illegally injected TN packets
- o Handling replay attacks (re-injecting previous TN messages)
- o TN messages propagating outside an operator's domain

Illegal TN packets can be detected with authentication check. Providing authentication for TN messages is described in [Section 6](#). Prevention of replay attacks needs authentication in combination with sequence numbering, which is also described at the same section.

Preventing TN messages that travel inline with data packets MUST be solved by nodes egressing the operator's domain. Solutions for IP and MPLS are described in sections [Section 8](#) and [Section 9](#), respectively.

## [13.](#) References

### [13.1.](#) Normative References

- [I-D.karan-mofrr]  
Karan, A., Filsfils, C., Farinacci, D., Decraene, B., Leymann, N., and W. Henderickx, "Multicast only Fast Re-Route", [draft-karan-mofrr-02](#) (work in progress),

Wijnands, et al. Expires July 28, 2014 [Page 20]

---

Internet-Draft Tree Notification for Multicast FRR January 2014

March 2012.

- [RFC5384] Boers, A., Wijnands, I., and E. Rosen, "The Protocol Independent Multicast (PIM) Join Attribute Format", [RFC 5384](#), November 2008.
- [RFC6388] Wijnands, IJ., Minei, I., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", [RFC 6388](#), November 2011.

### [13.2.](#) Informative References

[I-D.atlas-rtgwg-mrt-mc-arch]

Atlas, A., Kebler, R., Wijnands, I., Csaszar, A., and G. Envedi, "An Architecture for Multicast Protection Using Maximally Redundant Trees",  
[draft-atlas-rtgwg-mrt-mc-arch-02](#) (work in progress),  
July 2013.

#### Authors' Addresses

IJsbrand Wijnands (editor)  
Cisco  
De kleetlaan 6a  
Diegem, 1831  
Belgium

Phone:  
Email: [ice@cisco.com](mailto:ice@cisco.com)

Luc De Ghein  
Cisco  
De kleetlaan 6a  
Diegem, 1831  
Belgium

Phone:  
Email: [ldeghein@cisco.com](mailto:ldeghein@cisco.com)

Wijnands, et al.

Expires July 28, 2014

[Page 21]

---

Internet-Draft

Tree Notification for Multicast FRR

January 2014

Gabor Sandor Enyedi (editor)  
Ericsson  
Konyves Kalman Krt 11/B  
Budapest, 1097  
Hungary



Phone:  
Email: Gabor.Sandor.Enyedi@ericsson.com

Andras Csaszar  
Ericsson  
Konyves Kalman Krt 11/B  
Budapest, 1097  
Hungary

Phone:  
Email: Andras.Csaszar@ericsson.com

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

Email: Jeff.Tantsura@ericsson.com