TRILL working group
Internet Draft
Intended status: Standard Track
Expires: April 2015

L. Dunbar
D. Eastlake
Huawei
Radia Perlman
Intel
I. Gashinsky
Yahoo
October 7, 2014

Directory Assisted TRILL Encapsulation
draft-dunbar-trill-directory-assisted-encap-08.txt

Status of this Memo

Abstract

This draft describes how data center network can benefit from non-RBridge nodes performing TRILL encapsulation with assistance from directory service.

Table of Contents

1. Introduction

This draft describes how data center networks can benefit from non-RBridge nodes performing TRILL encapsulation with assistance from directory service.

[RFC7067] describes the framework for RBridge edge to get MAC&VLAN<->RBridgeEdge mapping from a directory service in data center environments instead of flooding unknown DAs across TRILL domain. If it has the needed directory information, any node, even a non-RBridge node, can perform the TRILL encapsulation. This draft is to describe the benefits and a scheme for non-RBridge nodes performing TRILL encapsulation.

2. Conventions used in this document

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].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

AF Appointed Forwarder RBridge port [RFC6439]

Bridge: IEEE 802.1Q compliant device. In this draft, Bridge is used interchangeably with Layer 2 switch.

DA: Destination Address

DC: Data Center

EoR: End of Row switches in data center. Also known as Aggregation switches in some data centers

Host: Application running on a physical server or a virtual machine. A host usually has at least one IP address and at least one MAC address.

SA: Source Address

ToR: Top of Rack Switch in data center. It is also known as access switches in some data centers.

TRILL-EN: TRILL Encapsulating node. It is a node that only performs the TRILL encapsulation but doesn't participate in RBridge's IS-IS routing.

VM: Virtual Machines

3. Directory Assistance to Non-RBridge

With directory assistance [RFC7067], a non-RBridge can be informed if a packet needs to be forwarded across the RBridge domain and the corresponding egress RBridge. Suppose the RBridge domain boundary starts at network switches (not virtual switches embedded on servers), a directory can assist Virtual Switches embedded on servers to encapsulate with a proper TRILL header by providing the nickname of the egress RBridge edge to which the destination is attached. The other information needed to encapsulate can be either learned by listening to TRILL Hellos, which will indicate the MAC address and nickname of appropriate edge RBridges, or by configuration.

If a destination is not attached to other RBridge edge nodes based on the directory [RFC7067], the non-RBridge node can forward the data frames natively, i.e. not encapsulating any TRILL header.

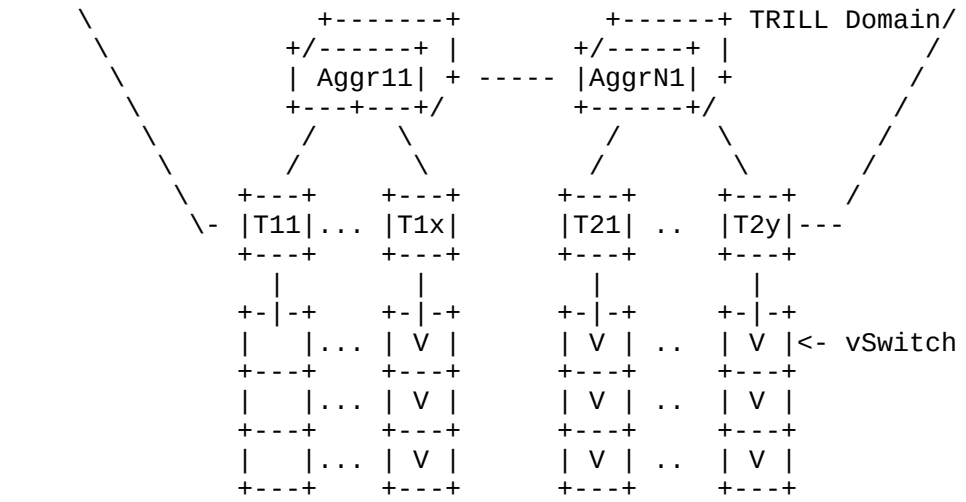


Figure 1 TRILL domain in typical Data Center Network

When a TRILL encapsulated data packet reaches the ingress RBridge, the ingress RBridge simply forwards the pre-encapsulated packet to the RBridge that is specified by the egress nickname field of the TRILL header of the data frame. When the ingress RBridge receives a native Ethernet frame, it handles it as usual and may drop it if it has complete directory information indicating that the target is not attached to the TRILL campus.

In this environment with complete directory information, the ingress RBridge doesn't flood or forward the received data frames when the DA in the Ethernet data frames is unknown.

When all attached nodes to ingress RBridge can pre-encapsulate TRILL header for traffic across the TRILL domain, the ingress RBridge don't need to encapsulate any native Ethernet frames to the TRILL domain. The attached nodes can be connected to multiple edge RBridges by having multiple ports or by an bridged LAN. Under this environment, there is no need to designate AF ports and all RBridge edge ports connected to one bridged LAN can receive and forward pre-encapsulated traffic, which can greatly improve the overall network utilization.

Note: [RFC6325] Section 4.6.2 Bullet 8 specifies that an RBridge port can be configured to accept TRILL encapsulated frames from a neighbor that is not an RBridge.

When a TRILL frame arrives at an RBridge whose nickname matches with the destination nickname in the TRILL header of the frame, the processing is exactly same as normal, i.e. the RBridge decapsulates the received TRILL frame and forwards the decapsulated frame to the target attached to its edge ports. When the DA of the decapsulated Ethernet frame is not in the egress RBridge's local MAC attachment tables, the egress RBridge floods the decapsulated frame to all attached links in the frame's VLAN, or drops the frame (if the egress RBridge is configured with the policy).

We call a node that only performs the TRILL encapsulation but doesn't participate in RBridge's IS-IS routing a TRILL Encapsulating node (TRILL-EN). The TRILL Encapsulating Node can get the MAC&VLAN<->RBridgeEdge mapping table pulled from directory servers [RFC7067].

Editor's note: RFC7067 has defined Push and Pull model for edge nodes to get directory mapping information. While Pull Model is relative simple for TRILL-EN to implement, Pushing requires some reliable flooding mechanism, like the one used by IS-IS, between the edge RBridge and the TRILL encapsulating node. Something like an extension to ES-IS might be needed.

Upon receiving a native Ethernet frame, the TRILL-EN checks the MAC&VLAN<->RBridgeEdge mapping table, and perform the corresponding TRILL encapsulation if the entry is found in the mapping table. If the destination address and VLAN of the received Ethernet frame doesn't exist in the mapping table and no positive reply from pulling request to a directory, the Ethernet frame is dropped or forwarded in native form to an edge RBridge.

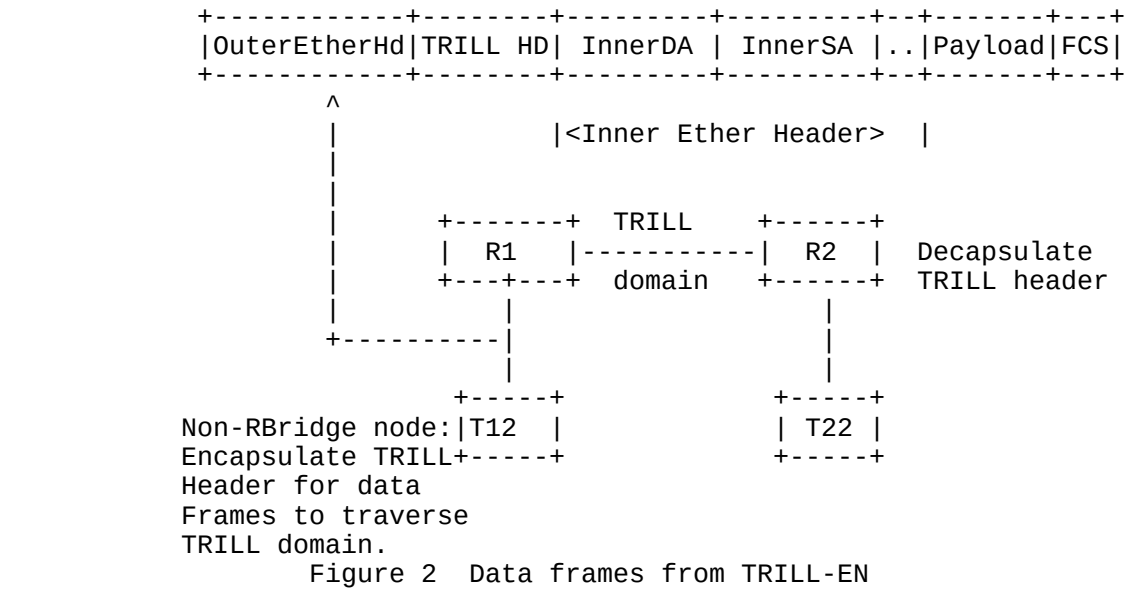


Figure 2 Data frames from TRILL-EN

4. Source Nickname in Frames Encapsulated by Non-RBridge Nodes

The TRILL header includes a Source RBridge's Nickname (ingress) and Destination RBridge's Nickname (egress). When a TRILL header is added by TRILL-EN, the Ingress RBridge edge node's nickname is used in the source address field.

5. Benefits of Non-RBridge encapsulating TRILL header

5.1. Avoid Nickname Exhaustion Issue

For a large Data Center with hundreds of thousands of virtualized servers, setting the TRILL boundary at the servers' virtual switches will create a TRILL domain with hundreds of thousands of RBridge nodes, which has issues of TRILL Nicknames exhaustion and challenges to IS-IS. On the other hand, setting TRILL boundary at aggregation switches that have many virtualized servers attached can limit the number of RBridge nodes in a TRILL domain, but introduce the issues of very large MAC&VLAN<->RBridgeEdge mapping table to be

maintained by RBridge edge nodes and the necessity of enforcing AF ports.

Allowing Non-RBridge nodes to pre-encapsulate data frames with TRILL header makes it possible to have a TRILL domain with a reasonable number of RBridge nodes in a large data center. All the TRILL-ENs attached to one RBridge are represented by one TRILL nickname, which can avoid the Nickname exhaustion problem.

5.2. Reduce MAC Tables for switches on Bridged LANs

When hosts in a VLAN (or subnet) span across multiple RBridge edge nodes and each RBridge edge has multiple VLANs enabled, the switches on the bridged LANs attached to the RBridge edge are exposed to all MAC addresses among all the VLANs enabled.

For example, for an Access switch with 40 physical servers attached, where each server has 100 VMs, there are 4000 hosts under the Access Switch. If indeed hosts/VMs can be moved anywhere, the worst case for the Access Switch is when all those 4000 VMs belong to different VLANs, i.e. the access switch has 4000 VLANs enabled. If each VLAN has 200 hosts, this access switch's MAC table potentially has 200*4000 = 800,000 entries.

If the virtual switches on servers pre-encapsulate the data frames destined for hosts attached to other RBridge Edge nodes, the outer MAC DA of those TRILL encapsulated data frames will be the MAC address of the local RBridge edge, i.e. the ingress RBridge. Therefore, the switches on the local bridged LAN don't need to keep the MAC entries for remote hosts attached to other edge RBridges.

But the traffic from nodes attached to other RBridges is decapsulated and has the true source and destination MACs. To prevent local bridges from learning remote hosts' MACs and adding to their MAC tables, one simple way is to disable this data plane learning on local bridges. The local bridges can be pre-configured with MAC addresses of local hosts with the assistance of a directory. The local bridges can always send frames with unknown Destination to the ingress RBridge. In an environment where a large number of VMs are instantiated in one server, the number of remote MAC addresses could be very large. If it is not feasible to disable learning and pre-configure MAC tables for local bridges, one effective method to minimize local bridges' MAC table size is to use the

server's MAC address to hide MAC addresses of the attached VMs. I.e. the server acting as an edge node using its own MAC address in the Source Address field of the packets originated from a host (or VM) embedded. When the Ethernet frame arrives at the target edge node (the server), the target edge node can send the packet to the corresponding destination host based on the packet's IP address. Very often, the target edge node communicates with the embedded VMs via a layer 2 virtual switch. Under this case, the target edge node can construct the proper Ethernet header with the assistance from directory. The information from directory includes the proper host IP to MAC mapping information.

6. Conclusion and Recommendation

When directory information is available, nodes outside the TRILL domain can encapsulate data frames destined for nodes attached to remote RBridges. The non-RBridge encapsulation approach is especially useful when there are a large number of servers in a data center equipped with hypervisor-based virtual switches. It is relatively easy for virtual switches, which are usually software based, to get directory assistance and perform network address encapsulation.

7. Manageability Considerations

It requires directory assistance to make it possible for a non-TRILL node to pre-encapsulate packets destined towards remote RBridges.

8. Security Considerations

Pull Directory queries and responses are transmitted as RBridge-to-RBridge or native RBridge Channel messages. Such messages can besecured as specified in [ChannelTunnel].

For general TRILL security considerations, see [RFC6325].

9. IANA Considerations

This document requires no IANA actions. RFC Editor: Please remove this section before publication.

10. References

10.1. Normative References

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

[RFC6325] Perlman, et, al, "Routing Bridges (RBridges): Base Protocol Specification", RFC6325, July 2011

[RFC6439] Perlman, R., Eastlake, D., Li, Y., Banerjee, A., and F. Hu, "Routing Bridges (RBridges): Appointed Forwarders", RFC 6439, November 2011.

10.2. Informative References

[RFC7067] Dunbar, et, al "Directory Assistance Problem and High-Level Design Proposal", RFC7067, Nov, 2013.

[ChannelTunnel] - D. Eastlake, Y. Li, "TRILL: RBridge Channel Tunnel Protocol", draft-eastlake-trill-channel-tunnel, work in progress.

11. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Linda Dunbar
Huawei Technologies
5340 Legacy Drive, Suite 175
Plano, TX 75024, USA
Phone: (469) 277 5840
Email: linda.dunbar@huawei.com

Donald Eastlake
Huawei Technologies
155 Beaver Street
Milford, MA 01757 USA
Phone: 1-508-333-2270
Email: d3e3e3@gmail.com

Radia Perlman
Intel Labs
2200 Mission College Blvd.
Santa Clara, CA 95054-1549 USA
Phone: 1-408-765-8080
Email: Radia@alum.mit.edu

Igor Gashinsky
Yahoo
45 West 18th Street 6th floor
New York, NY 10011
Email: igor@yahoo-inc.com

TRILL Working Group
INTERNET-DRAFT
Intended status: Proposed Standard
Obsoletes: 7180
Updates: 6325, 7177, 7179

Donald Eastlake
Mingui Zhang
Huawei
Radia Perlman
EMC
Ayan Banerjee
Cisco
Anoop Ghanwani
Dell
Sujay Gupta
IP Infusion

Expires: April 21, 2015
October 22, 2014

TRILL: Clarifications, Corrections, and Updates
<draft-eastlake-trill-rfc7180bis-01.txt>

Abstract

Since publication of the TRILL (Transparent Interconnection of Lots of Links) base protocol in 2011, active development of TRILL has revealed errata in RFC 6325 and areas that could use clarifications or updates. RFCs 7177, 7357, and [rfc6439bis] provide clarifications and updates with respect to Adjacency, the TRILL ESADI (End Station Address Distribution Information) protocol, and Appointed Forwarders respectively. This document provides other known clarifications, corrections, and updates. It obsoletes RFC 7180 (the previous TRILL clarifications, corrections), updates RFC 7177, updates RFC 7179, and updates RFC 6325.

Status of This Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Distribution of this document is unlimited. Comments should be sent to the TRILL working group mailing list: <trill@ietf.org>.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html. The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Table of Contents

Table of Contents (continued)

1. Introduction (Changed)

Since the TRILL base protocol [RFC6325] was published in 2011, active development of TRILL has revealed errors in the specification [RFC6325] and several areas that could use clarifications or updates.

[RFC7177], [RFC7357], and [rfc6439bis] provide clarifications and updates with respect to Adjacency, the TRILL ESADI (End Station Address Distribution Information) protocol, and Appointed Forwarders. This document provides other known clarifications, corrections, and updates to [RFC6325], [RFC7177], and [RFC7179]. This document obsoletes [RFC7180], the previous TRILL clarifications, corrections, and updates document.

Sections of this document are annotated as to whether they are "New" technical material, material that has been technically "Changed", or material that is technically "Unchanged" by the appearance of one of these three words in parenthesis at the end of the section header. A section with only editorial changes is annotated as "(Unchanged)". If no such notation appears, then the first notation encountered on going to successively higher-level headers applies. Appendix C describes changes, summarizes material added, and lists material deleted.

1.1 Precedence (Changed)

In case of conflict between this document and [RFC6325], [RFC7177], or [RFC7179] this document takes precedence. In addition, Section 1.2 (Normative Content and Precedence) of [RFC6325] is updated to provide a more complete precedence ordering of the sections of [RFC6325] as following, where sections to the left take precedence over sections to their right:

4 > 3 > 7 > 5 > 2 > 6 > 1

1.2 Changes That Are Not Backward Compatible (Unchanged)

The change made by Section 3.4 below, which was also present in [RFC7180], is not backward compatible with [RFC6325] but has nevertheless been adopted to reduce distribution tree changes resulting from topology changes.

The several other changes herein that are fixes to errata for [RFC6325] -- [Err3002] [Err3003] [Err3004] [Err3052] [Err3053] [Err3508] -- may not be backward compatible with previous implementations that conformed to errors in the specification.

1.3 Terminology and Acronyms (Changed)

This document uses the acronyms defined in [RFC6325], some of which are repeated below for convenience, along with some additional acronyms and terms as follows:

CFI - Canonical Format Indicator [802].

DEI - Drop Eligibility Indicator [802.1Q-2011].

EISS - Enhanced Internal Sublayer Service.

OOMF - Overload Originated Multi-destination Frame.

RBridge - An alternative name for a TRILL Switch.

RPFC - Reverse Path Forwarding Check.

SNPA - SubNetwork Point of Attachment (for example, MAC address).

TRILL - Transparent Interconnection of Lots of Links (or Tunneled Routing in the Link Layer).

TRILL Switch - A device implementing the TRILL protocol. An alternative name for an RBridge.

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

2. Overloaded and/or Unreachable RBridges (Unchanged)

In this Section 2, the term "neighbor" refers only to actual RBridges and ignores pseudonodes.

RBridges may be in overload as indicated by the [IS-IS] overload flag in their LSPs (Link State PDUs). This means that either (1) they are incapable of holding the entire link-state database and thus do not have a view of the entire topology or (2) they have been configured to have the overload bit set. Although networks should be engineered to avoid actual link-state overload, it might occur under various circumstances. For example, if a large campus included one or more low-end TRILL Switches.

It is a common operational practice to set the overload bit in an [IS-IS] router (such as a TRILL Switch) when performing maintenance on that router that might affect its ability to correctly forward packets; this will usually leave the router reachable for maintenance traffic, but transit traffic will not be routed through it. (Also, in some cases, TRILL provides for setting the overload bit in the pseudonode of a link to stop TRILL Data traffic on an access link (see Section 4.9.1 of [RFC6325]).)

[IS-IS] and TRILL make a reasonable effort to do what they can even if some TRILL Switches/routers are in overload. They can do reasonably well if a few scattered nodes are in overload. However, actual least-cost paths are no longer assured if any TRILL Switches are in overload.

For the effect of overload on the appointment of forwarders, see [rfc6439bis].

2.1 Reachability

Packets are not least-cost routed through an overloaded TRILL Switch, although they may originate or terminate at an overloaded TRILL Switch. In addition, packets will not be least-cost routed over links with cost 2**24 - 1 [RFC5305]; such links are reserved for traffic- engineered packets, the handling of which is beyond the scope of this document.

As a result, a portion of the campus may be unreachable for least-cost routed TRILL Data because all paths to it would be through either a link with cost 2**24 - 1 or through an overloaded RBridge. For example, an RBridge (TRILL Switch) RB1 is not reachable by TRILL Data if all of its neighbors are connected to RB1 by links with cost 2**24 - 1. Such RBridges are called "data unreachable".

The link-state database at an RBridge RB1 can also contain information on TRILL Switches that are unreachable by IS-IS link-state flooding due to link or RBridge failures. When such failures partition the campus, the TRILL Switches adjacent to the failure and on the same side of the failure as RB1 will update their LSPs to show the lack of connectivity, and RB1 will receive those updates. As a result, RB1 will be aware of the partition. Nodes on the far side of the partition are both IS-IS unreachable and data unreachable. However, LSPs held by RB1 for TRILL Switches on the far side of the failure will not be updated and may stay around until they time out, which could be tens of minutes or longer. (The default in [IS-IS] is twenty minutes.)

2.2 Distribution Trees

An RBridge in overload cannot be trusted to correctly calculate distribution trees or correctly perform the RPFC (Reverse-Path Forwarding Check). Therefore, it cannot be trusted to forward multi-destination TRILL Data packets. It can only appear as a leaf node in a TRILL multi-destination distribution tree. Furthermore, if all the immediate neighbors of an RBridge are overloaded, then it is omitted from all trees in the campus and is unreachable by multi-destination packets.

When an RBridge determines what nicknames to use as the roots of the distribution trees it calculates, it MUST ignore all nicknames held by TRILL Switches that are in overload or are data unreachable. When calculating RPFCs for multi-destination packets, an RBridge RB1 MAY, to avoid calculating unnecessary RPFC state, ignore any trees that cannot reach to RB1 even if other RBridges list those trees as trees that other TRILL Switches might use. (But see Section 3.)

2.3 Overloaded Receipt of TRILL Data Packets

The receipt of TRILL Data packets by overloaded RBridge RB2 is discussed in the subsections below. In all cases, the normal Hop Count decrement is performed, and the TRILL Data packets is discarded if the result is less than one or if the egress nickname is illegal.

2.3.1 Known Unicast Receipt

RB2 will not usually receive unicast TRILL Data packets unless it is the egress, in which case it egresses and delivers the data normally. If RB2 receives a unicast TRILL Data packet for which it is not the

egress, perhaps because a neighbor does not yet know it is in overload, RB2 MUST NOT discard the packet because the egress is an unknown nickname as it might not know about all nicknames due to its overloaded condition. If any neighbor, other than the neighbor from which it received the packet, is not overloaded, it MUST attempt to forward the packet to one of those neighbors selected at random [RFC4086]. If there is no such neighbor, the packet is discarded.

2.3.2 Multi-Destination Receipt

If RB2 in overload receives a multi-destination TRILL Data packet, RB2 MUST NOT apply an RPFC since, due to overload, it might not do so correctly. RB2 egresses and delivers the frame locally where it is Appointed Forwarder for the frame's VLAN, subject to any multicast pruning. But since, as stated above, RB2 can only be the leaf of a distribution tree, it MUST NOT forward a multi-destination TRILL Data packet (except as an egressed native frame where RB2 is Appointed Forwarder).

2.4 Overloaded Origination of TRILL Data Packets

Overloaded origination of unicast TRILL Data packets with known egress and of multi-destination packets is discussed in the subsections below.

2.4.1 Known Unicast Origination

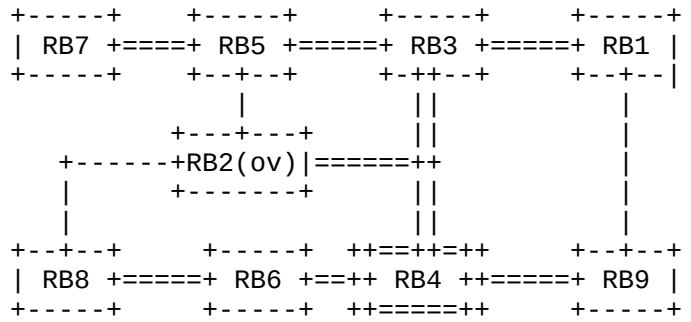
When an overloaded RBridge RB2 ingresses or creates a known destination unicast data packet, it delivers it locally if the destination is local. Otherwise, RB2 unicasts it to any neighbor TRILL Switch that is not overloaded. It MAY use what routing information it has to help select the neighbor.

2.4.2 Multi-Destination Origination

Overloaded RBridge RB2 ingressing or creating a multi-destination data packet is more complex than for the known unicast case as discussed below.

2.4.2.1 An Example Network

For example, consider the network below in which, for simplicity, end stations and any bridges are not shown. There is one distribution tree of which RB4 is the root, as represented by double lines. Only RBridge RB2 is overloaded.



Since RB2 is overloaded, it does not know what the distribution tree or trees are for the network. Thus, there is no way it can provide normal TRILL Data service for multi-destination native frames. So RB2 tunnels the frame to a neighbor that is not overloaded if it has such a neighbor that has signaled that it is willing to offer this service. RBridges indicate this in their Hellos as described below. This service is called OOMF (Overload Originated Multi- destination Frame) service.

- The multi-destination frame MUST NOT be locally distributed in native form at RB2 before tunneling to a neighbor because this would cause the frame to be delivered twice. For example, if RB2 locally distributed a multicast native frame and then tunneled it to RB5, RB2 would get a copy of the frame when RB3 transmitted it as a TRILL Data packet on the multi-access RB2-RB3-RB4 link. Since RB2 would, in general, not be able to tell that this was a frame it had tunneled for distribution, RB2 would decapsulate it and locally distribute it a second time.

- On the other hand, if there is no neighbor of RB2 offering RB2 the OOMF service, RB2 cannot tunnel the frame to a neighbor. In this case, RB2 MUST locally distribute the frame where it is Appointed Forwarder for the frame's VLAN and optionally subject to multicast pruning.

2.4.2.2 Indicating OOMF Support

An RBridge RB3 indicates its willingness to offer the OOMF service to RB2 in the TRILL Neighbor TLV in RB3's TRILL Hellos by setting a bit

associated with the SNPA (SubNetwork Point of Attachment, also known as MAC address) of RB2 on the link (see IANA Considerations). Overloaded RBridge RB2 can only distribute multi-destination TRILL Data packets to the campus if a neighbor of RB2 not in overload offers RB2 the OOMF service. If RB2 does not have OOMF service available to it, RB2 can still receive multi-destination packets from non-overloaded neighbors and, if RB2 should originate or ingress such a frame, it distributes it locally in native form.

2.4.2.3 Using OOMF Service

If RB2 sees this OOMF (Overload Originated Multi-destination Frame) service advertised for it by any of its neighbors on any link to which RB2 connects, it selects one such neighbor by a means beyond the scope of this document. Assuming RB2 selects RB3 to handle multi-destination packets it originates, RB2 MUST advertise in its LSP that it might use any of the distribution trees that RB3 advertises so that the RPFC will work in the rest of the campus. Thus, notwithstanding its overloaded state, RB2 MUST retain this information from RB3 LSPs, which it will receive as it is directly connected to RB3.

RB2 then encapsulates such frames as TRILL Data packets to RB3 as follows: M bit = 0, Hop Count = 2, ingress nickname = a nickname held by RB2, and, since RB2 cannot tell what distribution tree RB3 will use, egress nickname = a special nickname indicating an OOMF packet (see IANA Considerations). RB2 then unicasts this TRILL Data packet to RB3. (Implementation of Item 4 in Section 4 below provides reasonable assurance that, notwithstanding its overloaded state, the ingress nickname used by RB2 will be unique within at least the portion of the campus that is IS-IS reachable from RB2.)

On receipt of such a packet, RB3 does the following:

- changes the Egress Nickname field to designate a distribution tree that RB3 normally uses,
- sets the M bit to one,
- changes the Hop Count to the value it would normally use if it were the ingress, and
- forwards the packet on that tree.

RB3 MAY rate limit the number of packets for which it is providing this service by discarding some such packets from RB2. The provision of even limited bandwidth for OOMFs by RB3, perhaps via the slow path, may be important to the bootstrapping of services at RB2 or at end stations connected to RB2, such as supporting DHCP and ARP/ND (Address Resolution Protocol / Neighbor Discovery). (Everyone sometimes needs a little OOMF (pronounced "oomph") to get off the

ground.)

3. Distribution Trees and RPF Check (Changed)

Two corrections, a clarification, and two updates related to distribution trees appear in the subsections below along with an alternative, stronger RPF (Reverse Path Forwarding) Check. See also Section 2.2.

3.1 Number of Distribution Trees (Unchanged)

In [RFC6325], Section 4.5.2, page 56, Point 2, 4th paragraph, the parenthetical "(up to the maximum of {j,k})" is incorrect [Err3052]. It should read "(up to k if j is zero or the minimum of (j, k) if j is non-zero)".

3.2 Distribution Tree Update Clarification (Unchanged)

When a link-state database change causes a change in the distribution tree(s), there are several possibilities. If a tree root remains a tree root but the tree changes, then local forwarding and RPFC entries for that tree should be updated as soon as practical. Similarly, if a new nickname becomes a tree root, forwarding and RPFC entries for the new tree should be installed as soon as practical. However, if a nickname ceases to be a tree root and there is sufficient room in local tables, the forwarding and RPFC entries for the former tree MAY be retained so that any multi-destination TRILL Data packets already in flight on that tree have a higher probability of being delivered.

3.3 Multicast Pruning Based on IP Address (Unchanged)

The TRILL base protocol specification [RFC6325] provides for and recommends the pruning of multi-destination packet distribution trees based on the location of IP multicast routers and listeners; however, multicast listening is identified by derived MAC addresses as communicated in the Group MAC Address sub-TLV [RFC7176].

TRILL Switches MAY communicate multicast listeners and prune distribution trees based on the actual IPv4 or IPv6 multicast addresses involved. Additional Group Address sub-TLVs are provided in [RFC7176] to carry this information. A TRILL Switch that is only capable of pruning based on derived MAC address SHOULD calculate and use such derived MAC addresses from multicast listener IPv4/IPv6 address information it receives.

3.4 Numbering of Distribution Trees (Unchanged)

Section 4.5.1 of [RFC6325] specifies that, when building distribution tree number j, node (RBridge) N that has multiple possible parents in the tree is attached to possible parent number j mod p. Trees are numbered starting with 1, but possible parents are numbered starting with 0. As a result, if there are two trees and two possible parents, in tree 1, parent 1 will be selected, and in tree 2, parent 0 will be selected.

This is changed so that the selected parent MUST be (j-1) mod p. As a result, in the case above, tree 1 will select parent 0, and tree 2 will select parent 1. This change is not backward compatible with [RFC6325]. If all RBridges in a campus do not determine distribution trees in the same way, then for most topologies, the RPFC will drop many multi-destination packets before they have been properly delivered.

3.5 Link Cost Directionality (Unchanged)

Distribution tree construction, like other least-cost aspects of TRILL, works even if link costs are asymmetric, so the cost of the hop from RB1 to RB2 is different from the cost of the hop from RB2 to RB1. However, it is essential that all RBridges calculate the same distribution trees, and thus, all must either use the cost away from the tree root or the cost towards the tree root. As corrected in [Err3508], the text in Section 4.5.1 of [RFC6325] is incorrect. It says:

> In other words, the set of potential parents for N, for the tree rooted at R, consists of those that give equally minimal cost paths from N to R and ...

but the text should say "from R to N":

> In other words, the set of potential parents for N, for the tree rooted at R, consists of those that give equally minimal cost paths from R to N and ...

3.6 Alternative RPF Check (New)

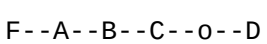
[RFC6325] mandates a Reverse Path Forwarding (RPF) Check on multi-destination TRILL data packets to avoid possible multiplication and/or looping of multi-destination traffic during TRILL campus topology transients. This check is logically performed at each TRILL switch input port and determines, based on where the packet started

(the ingress nickname) and the tree on which it is being distributed, whether it is arriving on the expected port. If not, the packet is silently discarded. This check is fine for point-to-point links; however, there are rare circumstances involving multi-access ("broadcast") links where a packet can be duplicated despite this RPF Check and other checks performed by TRILL.

Section 3.6.1 gives an example of the potential problem and Section 3.6.2 specifies a solution. This solution is an alternative stronger RPF Check that TRILL Switches can implemented in place of the RFF Check in [RFC6325].

3.6.1 Example of the Potential Problem

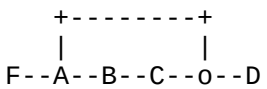
Consider this network:



All the links except the link between C and D are point-to-point links. C and D are connected over a broadcast link represented by the pseudonode "o". For example, C and D could be connected by a bridged LAN. (Bridged LANs are transparent to TRILL.)

Although the choice of root is unimportant here, assume that D or F is chosen as the root of a distribution tree so it is obvious the tree looks just like the diagram above.

Now assume a link comes up from A to the same bridged LAN. The network then looks like this:



Let’s say the resulting tree in steady state includes all links except the B-C link. After the network has converged, a packet that starts out from F will go F->A. Then A will send one copy on the A-B link and another copy into the bridge LAN from which it will be received by C and D.

Now consider a transition stage where A and D have acted on the new LSPs and programmed their forwarding plane, while B and C have not yet done so. This means that B and C both consider the link between them to still be part of the tree. In this case, a packet that starts out from F and reaches A will be copied by A into the A-B link and to the bridge LAN. D’s RPF check says to accept packets on this tree coming from F over its port on the bridged LAN, so it gets accepted. D is also adjacent to A on the tree, so the tree adjacency check, a

separate check mandated by [RFC6325] also passes.

However, the packet that gets to B gets sent out by B to C. C's RPF check still has the old state, and it thinks the packet is OK. C sends the packet along the old tree, which is into the bridge LAN. D receives one more packet, but the tree adjacency check passes at D because C is adjacent to D in the new tree as well. The RPF Check also passes at D because D's port on the bridged LAN is OK for receiving packets from F.

So, during this transient state, D gets duplicates of every multi-destination packet ingressed at F (unless the packet gets pruned) until B and C act on the new LSPs and program their hardware tables.

3.6.2 Solution and Discussion

The problem stems from the RPF Check in [RFC6325] depending only on the port at which a TRILL data packet is received, the ingress nickname, and the tree being used, that is, a check if {ingress nickname, tree, input port} is a valid combination according to the receiving TRILL switch's view of the campus topology. A multi-access link actually has multiple adjacencies overlaid on one physical link and to avoid the problem shown in Section 3.6.1, a stronger check is needed that includes the Layer 2 source address of the TRILL Data packet being received. (TRILL is a Layer 3 protocol and TRILL switches are true routers that logically strip the Layer 2 header from any arriving TRILL data packets and add the appropriate new Layer 2 header to any outgoing TRILL Data packet to get it to the next TRILL switch, so the Layer 2 source address in a TRILL Data packet identifies the immediately previous TRILL Switch that forwarded the packet.)

What is needed, instead of checking the validity of the triplet {ingress nickname, tree, input port} is to check that the quadruplet {ingress nickname, source SNPA, tree, input port} is valid (where "source SNPA" (Sub-Network Point of Access) is the Outer.MacSA for an Ethernet link). Although it is true that [RFC6325] also requires a check that a multi-destination TRILL Data packet is from a TRILL switch that is adjacent in the distribution tree being used, this is a separate check from the RPF Check and these two independent checks are not as powerful as the single unified check for a valid quadruplet.

However, this stronger RPF Check is not without cost. In the simple case of a multi-access link where each TRILL switch has only one port on the link, it merely increases the size of validity entries by adding the source SNPA (Outer.MacSA). However, assume some TRILL Switch RB1 has N ports attached to a multi-access link. RB1 is

permitted to load split multi-destination traffic it is sending into the multi-access link across those ports (Section 4.4.4 [RFC6325]). Assume RB2 is another TRILL Switch on the link and RB2 is distribution tree adjacent to RB1. The number of validity quadruplets at RB2 for ingress nicknames whose multi-destination traffic would arrive through RB1 is multiplied by N because RB2 has to accept such traffic from any of the ports RB1 has on the access-link. Although such instances seem to be very rare in practice, N could in principle be tens or even a hundred or more ports, vastly increasing the RPF check state at RB2 when this stronger RPF check is used.

Another potential cost of the stronger RPF Check is increased transient loss of multi-destination TRILL data packets during a topology change. For TRILL switch D, the new stronger RPF Check is (tree->A, Outer.MacSA=A, ingress=A, arrival port=if1) while the old one was (tree->A, Outer.MacSA=C, ingress=A, arrival port=if1). Suppose both A and B have switched to the new tree for multicast forwarding while D has not updated its RPF Check yet, then the multicast packet will be dropped at D’s if1. Since D still expects packet from "Outer.MacSA=C". But we do not have this packet loss issue if the weaker triplet check (tree->A, ingress=A, arrival port=if1) is used. Thus, the stronger check can increase the RPF Check discard of multi-destination packets during topology transients.

Because of these potential costs, implementation of this stronger RPF Check is optional; however, the TRILL protocol is updated to provide that TRILL Switches MUST, for multi-destination packets, either implement the RPF and other checks in [RFC6325] or implement this stronger RPF Check as a substitute for the [RFC6325] RPF and tree adjacency checks.

4. Nicknames Selection (Unchanged)

Nickname selection is covered by Section 3.7.3 of [RFC6325]. However, the following should be noted:

1. The second sentence in the second bullet item in Section 3.7.3 of [RFC6325] on page 25 is erroneous [Err3002] and is corrected as follows:

   - The occurrence of "IS-IS ID (LAN ID)" is replaced with "priority".
   - The occurrence of "IS-IS System ID" is replaced with "seven-byte IS-IS ID (LAN ID)".

   The resulting corrected sentence in [RFC6325] reads as follows:

   > "If RB1 chooses nickname x, and RB1 discovers, through receipt of an LSP for RB2 at any later time, that RB2 has also chosen x, then the RBridge or pseudonode with the numerically higher priority keeps the nickname, or if there is a tie in priority, the RBridge with the numerically higher seven-byte IS-IS ID (LAN ID) keeps the nickname, and the other RBridge MUST select a new nickname."

2. In examining the link-state database for nickname conflicts, nicknames held by IS-IS unreachable TRILL Switches MUST be ignored, but nicknames held by IS-IS reachable TRILL Switches MUST NOT be ignored even if they are data unreachable.

3. An RBridge may need to select a new nickname, either initially because it has none or because of a conflict. When doing so, the RBridge MUST consider as available all nicknames that do not appear in its link-state database or that appear to be held by IS-IS unreachable TRILL Switches; however, it SHOULD give preference to selecting new nicknames that do not appear to be held by any TRILL Switch in the campus, reachable or unreachable, so as to minimize conflicts if IS-IS unreachable TRILL Switches later become reachable.

4. An RBridge, even after it has acquired a nickname for which there appears to be no conflicting claimant, MUST continue to monitor for conflicts with the nickname or nicknames it holds. It does so by checking in LSP PDUs it receives that should update its link-state database for the following: any occurrence of any of its nicknames held with higher priority by some other TRILL Switch that is IS-IS reachable from it. If it finds such a conflict, it MUST select a new nickname, even when in overloaded state. (It is possible to receive an LSP that should update the link-state database but does not do so due to overload.)

5. In the very unlikely case that an RBridge is unable to obtain a nickname because all valid RBridge nicknames (0x0001 through 0xFFBF inclusive) are in use with higher priority by IS-IS reachable TRILL Switches, it will be unable to act as an ingress, egress, or tree root but will still be able to function as a transit TRILL Switch. Although it cannot be a tree root, such an RBridge is included in distribution trees computed for the campus unless all its neighbors are overloaded. It would not be possible to send a unicast RBridge Channel message specifically to such a TRILL Switch [RFC7178]; however, it will receive unicast RBridge Channel messages sent by a neighbor to the Any-RBridge egress nickname and will receive appropriate multi-destination RBridge Channel messages.

5. MTU (Maximum Transmission Unit) (Unchanged)

MTU values in TRILL key off the originatingL1LSPBufferSize value communicated in the IS-IS originatingLSPBufferSize TLV [IS-IS]. The campus-wide value Sz, as described in Section 4.3.1 of [RFC6325], is the minimum value of originatingL1LSPBufferSize for the RBridges in a campus, but not less than 1470. The MTU testing mechanism and limiting LSPs to Sz assures that the LSPs can be flooded by IS-IS and thus that IS-IS can operate properly.

If nothing is known about the MTU of the links or the originatingL1LSPBufferSize of other RBridges in a campus, the originatingL1LSPBufferSize for an RBridge should default to the minimum of the LSP size that its TRILL IS-IS software can handle and the minimum MTU of the ports that it might use to receive or transmit LSPs. If an RBridge does have knowledge of link MTUs or other RBridge originatingL1LSPBufferSize, then, to avoid the necessity to regenerate the local LSPs using a different maximum size, the RBridge's originatingL1LSPBufferSize SHOULD be configured to the minimum of (1) the smallest value that other RBridges are or will be announcing as their originatingL1LSPBufferSize and (2) a value small enough that the campus will not partition due to a significant number of links with limited MTU. However, as provided in [RFC6325], in no case can originatingL1LSPBufferSize be less than 1470. In a well-configured campus, to minimize any LSP regeneration due to re-sizing, all RBridges will be configured with the same originatingL1LSPBufferSize.

Section 5.1 below corrects errata in [RFC6325], and Section 5.2 clarifies the meaning of various MTU limits for TRILL Ethernet links.

5.1 MTU-Related Errata in RFC 6325

Three MTU-related errata in [RFC6325] are corrected in the subsections below.

5.1.1 MTU PDU Addressing

Section 4.3.2 of [RFC6325] incorrectly states that multi-destination MTU-probe and MTU-ack TRILL IS-IS PDUs are sent on Ethernet links with the All-RBridges multicast address as the Outer.MacDA [Err3004]. As TRILL IS-IS PDUs, when multicast on an Ethernet link, they MUST be sent to the All-IS-IS-RBridges multicast address.

5.1.2 MTU PDU Processing

As discussed in [RFC6325] and, in more detail, in [RFC7177], MTU-probe and MTU-ack PDUs MAY be unicast; however, Section 4.6 of [RFC6325] erroneously does not allow for this possibility [Err3003]. It is corrected by replacing Item numbered "1" in Section 4.6.2 of [RFC6325] with the following quoted text to which TRILL Switches MUST conform:

"1. If the Ethertype is L2-IS-IS and the Outer.MacDA is either All-IS-IS-RBridges or the unicast MAC address of the receiving RBridge port, the frame is handled as described in Section 4.6.2.1"

The reference to "Section 4.6.2.1" in the above quoted text is to that section in [RFC6325].

5.1.3 MTU Testing

The last two sentences of Section 4.3.2 of [RFC6325] have errors [Err3053]. They currently read:

> "If X is not greater than Sz, then RB1 sets the "failed minimum MTU test" flag for RB2 in RB1's Hello. If size X succeeds, and X > Sz, then RB1 advertises the largest tested X for each adjacency in the TRILL Hellos RB1 sends on that link, and RB1 MAY advertise X as an attribute of the link to RB2 in RB1's LSP."

They should read:

> "If X is not greater than or equal to Sz, then RB1 sets the "failed minimum MTU test" flag for RB2 in RB1's Hello. If size X succeeds, and X >= Sz, then RB1 advertises the largest tested X for each adjacency in the TRILL Hellos RB1 sends on that link, and RB1 MAY advertise X as an attribute of the link to RB2 in RB1's LSP."

5.2 Ethernet MTU Values

originatingL1LSPBufferSize is the maximum permitted size of LSPs starting with the 0x83 Intradomain Routeing Protocol Discriminator byte. In Layer 3 IS-IS, originatingL1LSPBufferSize defaults to 1492 bytes. (This is because, in its previous life as DECnet Phase V, IS-IS was encoded using the SNAP SAP (Sub-Network Access Protocol Service Access Point) [RFC7042] format, which takes 8 bytes of overhead and 1492 + 8 = 1500, the classic Ethernet maximum. When

standardized by ISO/IEC [IS-IS] to use Logical Link Control (LLC) encoding, this default could have been increased by a few bytes but was not.)

In TRILL, originatingL1LSPBufferSize defaults to 1470 bytes. This allows 27 bytes of headroom or safety margin to accommodate legacy devices with the classic Ethernet maximum MTU despite headers such as an Outer.VLAN.

Assuming the campus-wide minimum link MTU is Sz, RBridges on Ethernet links MUST limit most TRILL IS-IS PDUs so that PDUz (the length of the PDU starting just after the L2-IS-IS Ethertype and ending just before the Ethernet Frame Check Sequence (FCS)) does not to exceed Sz. The PDU exceptions are TRILL Hello PDUs, which MUST NOT exceed 1470 bytes, and MTU-probe and MTU-ack PDUs that are padded by an amount that depends on the size being tested (which may exceed Sz).

Sz does not limit TRILL Data packets. They are only limited by the MTU of the devices and links that they actually pass through; however, links that can accommodate IS-IS PDUs up to Sz would accommodate, with a generous safety margin, TRILL Data packet payloads of (Sz - 24) bytes, starting after the Inner.VLAN and ending just before the FCS.

Most modern Ethernet equipment has ample headroom for frames with extensive headers and is sometimes engineered to accommodate 9K byte jumbo frames.

6. TRILL Port Modes (Unchanged)

Section 4.9.1 of [RFC6325] specifies four mode bits for RBridge ports but may not be completely clear on the effects of various combinations of bits.

The table below explicitly indicates the effect of all possible combinations of the TRILL port mode bits. "*" in one of the first four columns indicates that the bit can be either zero or one. The following columns indicate allowed frame types. The Disable bit normally disables all frames, but, as an implementation choice, some or all low-level Layer 2 control message can still be sent or received. Examples of Layer 2 control messages are those control frames for Ethernet identified in Section 1.4 of [RFC6325] or PPP link negotiation messages [RFC6361].

Disable	Access	Trunk	Layer 2 Control	native egress	native ingress, TRILL Data	LSP, SNP, MTU	TRILL Hello	P2P Hello
0	0	0	Yes	Yes	Yes	Yes	Yes	No
0	0	1	Yes	No	Yes	Yes	Yes	No
0	1	0	Yes	Yes	No	Yes	No	No
0	1	1	Yes	No	No	Yes	No	No
1	*	*	Optional	No	No	No	No	No
0	0	0	Yes	Yes	Yes	Yes	No	Yes
0	0	1	Yes	No	Yes	Yes	No	Yes
0	1	0	Yes	Yes	No	No	No	Yes
0	1	1	Yes	No	No	No	No	Yes

The formal name of the "access bit" above is the "TRILL traffic disable bit". The formal name of the "trunk bit" is the "end-station service disable bit" [RFC6325].

7. The CFI/DEI Bit (Unchanged)

In May 2011, the IEEE promulgated [802.1Q-2011], which changed the meaning of the bit between the priority and VLAN ID bits in the payload of C-VLAN tags. Previously, this bit was called the CFI (Canonical Format Indicator) bit [802] and had a special meaning in connection with IEEE 802.5 (Token Ring) frames. Now, under [802.1Q-2011], it is a DEI (Drop Eligibility Indicator) bit, similar to that bit in S-VLAN/B-VLAN tags where this bit has always been a DEI bit.

The TRILL base protocol specification [RFC6325] assumed, in effect, that the link by which end stations are connected to TRILL Switches and the restricted virtual link provided by the TRILL Data packet are IEEE 802.3 Ethernet links on which the CFI bit is always zero. Should an end station be attached by some other type of link, such as a Token Ring link, [RFC6325] implicitly assumed that such frames would be canonicalized to 802.3 frames before being ingressed, and similarly, on egress, such frames would be converted from 802.3 to the appropriate frame type for the link. Thus, [RFC6325] required that the CFI bit in the Inner.VLAN, which is shown as the "C" bit in Section 4.1.1 of [RFC6325], always be zero.

However, for TRILL Switches with ports conforming to the change incorporated in the IEEE 802.1Q-2011 standard, the bit in the Inner.VLAN, now a DEI bit, MUST be set to the DEI value provided by the EISS (Enhanced Internal Sublayer Service) interface on ingressing a native frame. Similarly, this bit MUST be provided to the EISS when transiting or egressing a TRILL Data packet. As with the 3-bit Priority field, the DEI bit to use in forwarding a transit packet MUST be taken from the Inner.VLAN. The exact effect on the Outer.VLAN DEI and priority bits and whether or not an Outer.VLAN appears at all on the wire for output frames may depend on output port configuration.

TRILL campuses with a mixture of ports, some compliant with [802.1Q-2011] and some compliant with pre-802.1Q-2011 standards, especially if they have actual Token Ring links, may operate incorrectly and may corrupt data, just as a bridged LAN with such mixed ports and links would.

8. Other IS-IS Considerations (Changed)

This section covers E-L1FS Support, Control Packet Priorities, Unknown PDUs, the Nickname Flags APPsub-TLV, and Graceful Restart.

8.1 E-L1FS Support (New)

TRILL switches MUST support Extended Level 1 Flooding Scope PDUs (E-L1FS) [RFC7356] and MUST include a Scoped Flooding Support TLV [RFC7356] in all TRILL Hellos they send indicating support for this scope and any other FS-LSP scopes that they support. This support increases the number of fragments available for link state information by over two orders of magnitude. (See Section 9 for further information on support of the Scoped Flooding Support TLV.)

In addition, TRILL switches MUST advertise their support of E-L1FS flooding in a TRILL Version sub-TLV capability bit (see [RFC7176] and Section 11.2). This bit is used by a TRILL switch, say RB1, to determine support for E-L1FS by some remote RBx. The alternative of simply looking for an E-L1FS FS-LSP originated by RBx fails because (1) RBx might support E-L1FS flooding but not be originating any E-L1FS FS-LSPs and (2) even if RBx is originating E-L1FS FS-LSPs there might, due to legacy TRILL switches in the campus, be no path between RBx and RB1 through TRILL switches supporting E-L1FS flooding. If that were the case, no E-L1FS FS-LSP originated by RBx could get to RB1.

8.1.1 Backward Compatibility

A TRILL campus might contain TRILL switches supporting E-L1FS flooding and legacy TRILL switches that do not support E-L1FS or perhaps do not support any [RFC7356] scopes.

A TRILL switch conformant to this document can always tell which adjacent TRILL switches support E-L1FS flooding from the adjacency table entries on its ports (see Section 9). In addition, such a TRILL switch can tell which remote TRILL switches in a campus support E-L1FS by the presence of a TRILL Version sub-TLV in that TRILL switch's LSP with the E-L1FS support bit set in the Capabilities field; this capability bit is ignored for adjacent TRILL switches for which only the adjacency table entry is consulted to determine E-L1FS support.

TRILL specifications making use of E-L1FS MUST specify how situations involving mixed TRILL campus of TRILL switches will be handled.

8.1.2 E-L1FS Use for Existing (sub)TLVs

In a campus where all TRILL switches support E-L1FS, all TRILL sub-TLVs listed in Section 2.3 of [RFC7176], except the TRILL Version sub-TLV, MAY be advertised by inclusion in Router Capability or MT-Capability TLVs in E-L1FS FS-LSPs [RFC7356]. (The TRILL Version sub-TLV still MUST appear in an LSP fragment zero.)

In a mixed campus where some TRILL switches support E-L1FS and some do not, then only the following four sub-TLVs of those listed in Section 2.3 of [RFC7176] can appear in E-L1FS and then only under the conditions discussed below. In the following list, each sub-TLV is preceded by an abbreviated acronym used only in this Section 8.1.2:

IV: Interested VLANs and Spanning Tree Roots sub-TLV
VG: VLAN Groups sub-TLV
IL: Interested Labels and Spanning Tree Roots sub-TLV
LG: Label Groups sub-TLV

An IV or VG sub-TLV MUST NOT be advertised by TRILL switch RB1 in an E-L1FS FS-LSP and MUST be advertised in an LSP unless the following conditions are met:

- E-L1FS is supported by all of the TRILL switches that are data reachable from RB1 and are interested in the VLANs mentioned in the IV or VG sub-TLV, and
- there is E-L1FS connectivity between all such TRILL switches in the campus interested in the VLANs mentioned in the IV or VG sub-TLV (connectivity involving only intermediate TRILL switches that also support E-L1FS).

Any IV and VG sub-TLVs MAY still be advertised via core TRILL IS-IS LSP by any TRILL switch that has enough room in its LSPs.

The conditions for using E-L1FS for the IL and LG sub-TLVs are the same as for IV and VG but with Fine Grained Labels [RFC7172] substituted for VLANs.

Note, for example, that the above would permit a contiguous subset of the campus that supported Fine Grained Labels and E-L1FS to use E-L1FS to advertise IL and LG sub-TLVs even if the remainder of the campus did not support Fine Grained Labels or E-L1FS.

8.2 Control Packet Priorities (New)

When deciding what packet to send out a port, control packets used to establish and maintain adjacency between TRILL switches SHOULD be treated as being in the highest priority category. This includes TRILL IS-IS Hello and MTU PDUs and possibly other adjacency [RFC7177]

or link technology specific packets. Other control and data packets SHOULD be given lower priority so that a flood of such other packets cannot lead to loss of or inability to establish adjacency. Loss of adjacency causes a topology transient that can result in reduced throughput, reordering, increased probability of loss of multi-destination data, and, if the adjacency is a cut point, network partitioning.

Other important control packets should be given second highest priority. Lower priorities should be given to data or less important control packets.

Control packets can be ordered into priority classes as shown below. Although few implementations will actually treat all of these classes differently, higher numbered classes SHOULD NOT be treated as higher priority than lower numbered class. There may be additional control packets, not specifically listed in any category below, that SHOULD be handled as being in the most nearly analogous category.

1. Hello, MTU-probe, MTU-ack, and other packets critical to establishing and maintaining adjacency.

2. LSPs, CSNP/PSNPs, and other important control packets,

3. Circuit scoped FS-LSP, FS-CSNP, and FS-PSNPs.

4. Non-circuit scoped FS-LSP, FS-CSNP, and FS-PSNPs.

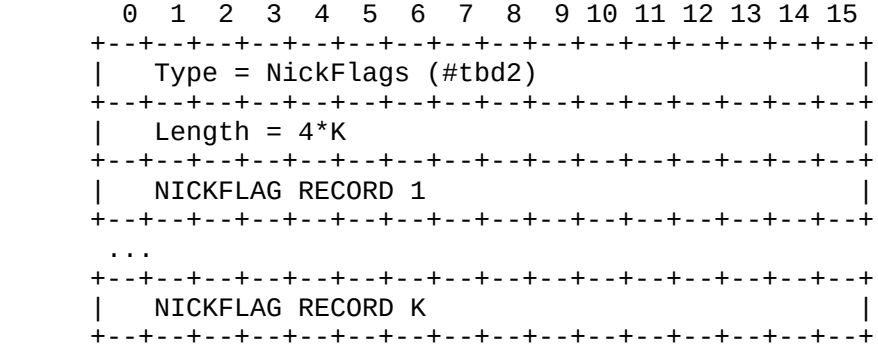
8.3 Unknown PDUs (New)

TRILL switches MUST silently discard [IS-IS] PDUs they receive with PDU numbers they do not understand, just as they ignore TLVs and sub-TLVs they receive that have unknown Types and sub-Types; however, they SHOULD maintain a counter of how many such PDUs have been received, on a per PDU number basis. (This is not burdensome as the PDU number is only a 5-bit field.)

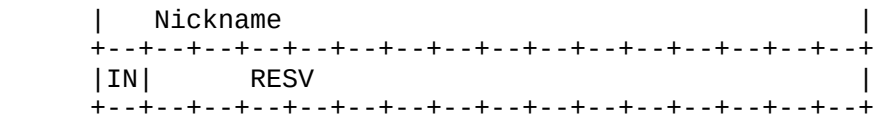
Note: The set of valid [IS-IS] PDUs was stable for so long that some IS-IS implementations may treat PDUs with unknown PDU numbers as a serious error and, for example, an indication that other valid PDUs from the sender are not to be trusted or that they should drop adjacency to the sender if it was adjacent. However, the MTU-probe and MTU-ack PDUs were added by [RFC7176] and now [RFC7356] has added three more new PDUs. While the authors of this document are not aware of any Internet drafts calling for further PDUs, the eventual addition of further new PDUs should not be surprising.

8.4 Nickname Flags APPsub-TLV (New)

An optional Nickname Flags APPsub-TLV within the TRILL GENINFO TLV [RFC7357] is specified below.



Where each NICKFLAG RECORD has the following format:



- Type: NickFlags TRILL APPsub-TLV, set to tbd2 (NICKFLAGS)
- Length: 4 times the number of NICKFLAG RECORDS present.
- Nickname: A 16-bit TRILL nickname held by the advertising TRILL switch ([RFC6325] and Section 4).
- IN: Ingress. If this flag is one, it indicates the advertising TRILL switch may use the nickname in the NICKFLAG RECORD as the ingress nickname of TRILL Headers it creates. If the flag is zero, that nickname will not be used for that purpose.
- RESV: Reserved for additional flags to be specified in the future. MUST be sent as zero and ignored on receipt.

A NICKFLAG RECORD is ignored if the nickname it lists is not a nickname owned by the TRILL switch advertising the enclosing NickFlags APPsub-TLV.

If a TRILL switch intends to use a nickname in the ingress nickname field of TRILL Headers it constructs, it can advertise this through E-L1FS FS-LSPs (see Section 8.1) using a NickFlags APPsub-TLV entry with the IN flag set. If it owns only one nickname, there is no reason to do this because, if a TRILL switch advertises no NickFlags

APPsub-TLVs with the IN flag set for nicknames it owns, it is assumed that the TRILL switch might use any or all nicknames it owns as the ingress nickname in TRILL Headers it constructs.

Every reasonable effort should be made to be sure that Nickname sub-TLVs [RFC7176] and NickFlags APPsub-TLVs remain in sync. If all TRILL switches in a campus support E-L1FS, so that Nickname sub-TLVs can be advertised in E-L1FS FS-LSPs, then the Nickname sub-TLV and any NickFlags APP-subTLVs for any particular nickname should be advertised in the same fragment. If they are not in the same fragment then, to the extent practical, all fragments involving those sub-TLVs for the same nickname should be propagated as an atomic action. If a TRILL switch sees multiple NickFlags APPsub-TLV entries for the same nickname, it assumes that nickname might be used as the ingress in a TRILL Header if any of the NickFlags APPsub-TLV entries have the IN bit set.

It is possible that a NickFlags APPsub-TLV would not be propagated throughout the TRILL campus due to legacy TRILL switches not supporting E-L1FS. In that case, Nickname sub-TLVs must be advertised in LSPs and TRILL switches not receiving NickFlags APPsub-TLVs having entries with the IN flag set will simply assume that the source TRILL switch might use any of its nicknames as ingress in constructing TRILL Headers. Thus the use of this optional APPsub-TLV is backwards compatible with legacy lack of E-L!FS support.

Additional flags may be assigned for other purposes out of the RESV field for other purposes in the future.

8.5 Graceful Restart (Unchanged)

TRILL Switches SHOULD support the features specified in [RFC5306], which describes a mechanism for a restarting IS-IS router to signal to its neighbors that it is restarting, allowing them to reestablish their adjacencies without cycling through the down state, while still correctly initiating link-state database synchronization. If this feature is not supported, it may increase the number of topology transients cause by a TRILL switch rebooting due to errors or maintenance.

9. Updates to [RFC7177] (Adjacency) [Changed]

To support the E-L1FS flooding scope [RFC7356] mandated by Section 8.1 and backwards compatibility with legacy RBridges not supporting E-L1FS flooding, the following updates are made to [RFC7177]:

1. The list in the second paragraph of [RFC7177] Section 3.1 has the following item added:

   - The Scoped Flooding Support TLV.

   In addition, the sentence immediately after that list is modified to read as follows:

   Of course, the priority, Desired Designated VLAN, Scoped Flooding Support TLV, and possibly the inclusion or value of the PORT-TRILL-VER sub-TLV, and/or BFD-Enabled TLV can change on occasion, but then the new value(s) must similarly be used in all TRILL Hellos on the LAN port, regardless of VLAN.

2. An additional bullet item is added to the end of Section 3.2 of [RFC7177] as follows:

   o The value from the Scoped Flooding Support TLV or a null string if none was included.

3. Near the bottom of Section 3.3 of [RFC7177] a bullet item as follows is added:

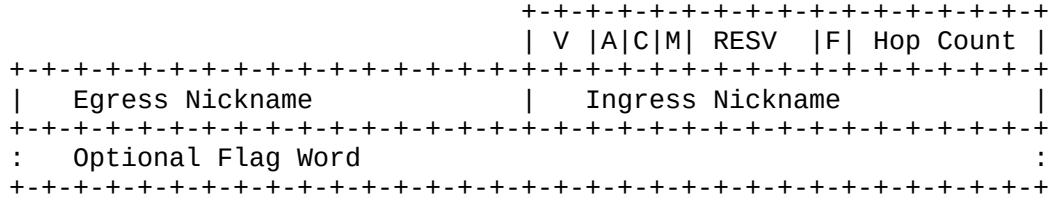
   o The variable length value part of the Scoped Flooding Support TLV in the Hello or a null string if that TLV does not occur in the Hello.

4. At the beginning of Section 4 of [RFC7177], a bullet item is added to the list as follows:

   o The variable length value part of the Scoped Flooding Support TLV used in TRILL Hellos sent on the port.

10. TRILL Header Update (New)

The TRILL header has been updated from its original specification in [RFC6325] by [TRILL-OAM-FM] and [RFC7179] and is further updated by this document. The TRILL header is now as show below and is followed by references for all of the fields. Those fields for which the reference is only to [RFC6325] are unchanged from that RFC.



In calculating a TRILL data packet hash as part of equal-cost multi-path selection, a TRILL switch MUST ignore the value of the "A" and "C" bits. In [RFC6325] and [RFC7179] the RESV and F fields above together constituted the "Ex-Length" or TRILL Header Extensions Length field.

- V (Version): 2-bit unsigned integer. See Section 3.2 of [RFC6325].
- A (Alert): 1 bit. See [TRILL-OAM-FM].
- C (Color): 1 bit. See Section 10.1.
- M (Multi-destination): 1 bit. See Section 3.4 of [RFC6325].
- RESV: 4 bits. These bits are reserved and MUST be sent as zero. They SHOULD be ignored on receipt; however, due to their previous use as specified in [RFC6325], some TRILL fast path harware implementations trap and do not forward TRILL Data packets with these bits non-zero.
- F: 1 bit. If this field is non-zero, then the optional Flag Word described in Section 10.2 is present. If it is zero, the Flag Word is not prsent.
- Hop Count: 6 bits. See Section 3.6 of [RFC6325] and Section 10.2.1 below.
- Egress Nickname. See Section 3.7.1 of [RFC6325].
- Ingress Nickname. See Section 3.7.2 of [RFC6325].
- Optional Flag Word: See [RFC7179] and Section 10.2.

10.1 Color Bit

The Color bit provides an optional way by which ingress TRILL switches MAY mark TRILL Data packets for implementation specific purposes. Transit TRILL switches MUST NOT change this bit. Transit and egress TRILL switches MAY use the Color bit for implementation dependent traffic labeling or statistical or other traffic study or analysis.

10.2 Flag Word Changes (update to [RFC7179])

When the extension length field is non-zero, the first 32 bits after the Ingress nickname field provides additional flags. These bits are as specified in [RFC7179] except as changed by the subsections below that provide extended Hop Count and extended Color fields. See Section 10.3 for a diagram and summary of these fields.

10.2.1 Extended Hop Count

The TRILL base protocol [RFC6325] specifies the Hop Count field in the header, to avoid packets persisting in the network due to looping or the like. However, the Hop Count field size (6 bits) limits the maximum hops a TRILL data packet can traverse to 64. Optionally, TRILL switches can use a field composed of bits 14 through 16 in the Flag Word, as specified below. to extend this field to 9 bits. This increases the maximum Hop Count to 512. Use of Hop Counts in excess of 64 requires support of this optional capability at all TRILL switches along the path of a TRILL Data packet.

10.2.1.1 Advertising Support

In case of a TRILL campus such that the unicast calculated path, plus a reasonable allowance for alternate pathing, or the distribution tree calculated path, traverse more than 64 hops, it may be that not all the TRILL switches support the extended Hop Count mechanism. As such it is required that TRILL switches advertise their support by setting bit 14 in the TRILL Version Sub-TLV Capabilities and Header Flags Supported field [RFC7176]; bits 15 and 16 of that field are now specified as Unassigned (see Section 11.2.5).

10.2.1.2 Ingress Behavior

If an ingress TRILL switch determines it should set the hop count for a TRILL Data packet to 63 or less, then behavior is as specified in the TRILL base protocol [RFC6325]. If hop count for a TRILL Data packet should be set to some value greater than 63 but less than 512 and all TRILL switches that the packet is reasonably likely to encounter support extended Hop Count, then the resulting TRILL Header has the Flag Word extension present, the high order three bits of the desired hop count are stored in the extended Hop Count field in the Flag Word, the five low order bits are stored in the Hop Count filed in the first word of the TRILL Header, and bit two (the Critical Reserved bit of the Critical Summary Bits) in the Flag Word is set.

For known unicast traffic (TRILL Header M bit zero), when an ingress TRILL switch determines that the least cost path to the egress is more than 64 hops but not all TRILL switches on that path support the extended Hop Count feature, the frame is discarded.

For multi-destination traffic, when a TRILL switch determines that one or more tree path from the ingress it more than 64 hops but not all TRILL switches in the campus support the extended Hop Count feature, the encapsulation uses a total Hop Count of 63 to obtain at least partial distribution of the traffic.

10.2.1.3 Transit Behavior

A transit TRILL switch supporting extended Hop Count behaves like a base protocol [RFC6325] TRILL switch in decrementing the hop count except that it considers the hop count to be a 9 bit file where the extended Hop Count field consistutes the high order three bits.

To be more precise: a TRILL switch supporting extended Hop Count takes the first of the following actions that is applicable:

1. If both the Hop Count and extended Hop Count fields are zero, the packet is discarded.

2. If the Hop Count is non-zero, it is decremented. As long as the extended Hop Count is non-zero, no special action is taken if the result of this decrement is zero and the packet is processed normally.

3. If the Hop Count is zero, it is set to the maximum value of 63 and the extended Hop Count is decremented.

10.2.1.4 Egress Behavior

No special behavior is required when egressing a TRILL Data packet that uses the extended Hop Count. The Flag Word, if present, is removed along with the rest of the TRILL Header during decapsulation.

10.2.2 Extended Color Field

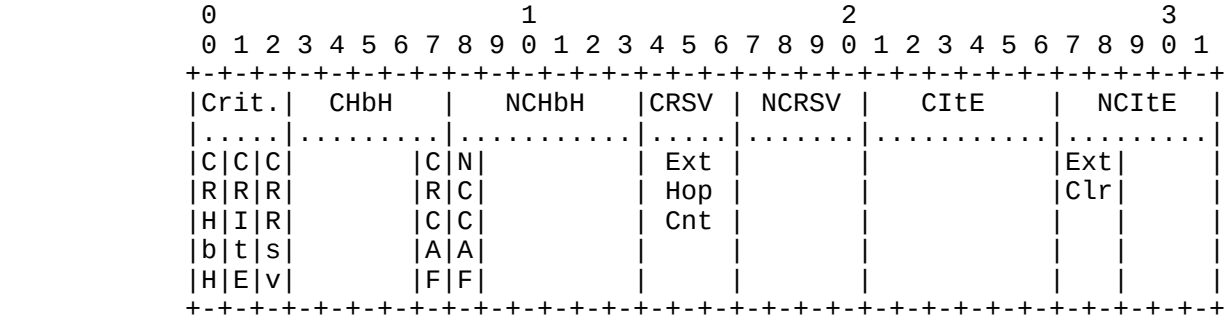
Flag Word bits 27 and 28 are specified to be a two-bit Extended Color field (see Section 10.3). These bits are in the non-critical ingress-to-egress region of the Flag Word.

The Extended Color field provides an optional way by which ingress TRILL switches MAY mark TRILL Data packets for implementation specific purposes. Transit TRILL switches MUST NOT change this bit. Transit and egress TRILL switches MAY use the Color bit for implementation dependent traffic labeling or statistical or other traffic study or analysis.

As provided in Section 2.3.1 of [RFC7176], support for these bits is indicated by the same bits (27 and 28) in the Capabilities and Header Flags Supported field of the TRILL Version Sub-TLV. In the spirit of indicating support, a TRILL switch that sets or senses the Extended Color field SHOULD set the corresponding 2-bit field in the TRILL Version Sub-TLV non-zero. The meaning of the possible non-zero values (1, 2 or 3) is implementation dependent.

10.3 Updated Flag Word Summary

With the changes above, the 32-bit Flag Word extension to the TRILL Header [RFC7179], appearing as the "TRILL Extended Header Flags" registry on the TRILL Parameters IANA web page, is now as follows:



Bit 0 to 2 are the Critical Summary bits as specified in [RFC7179]

consisting of the Critical Hop-by-Hop, Critical Ingres-to-Egress, and Critical Reserved bits, respectively. The next two fields are specific Critical and Non-Critical Hop-by-Hop bits, CHbH and NCHbH, respectively, containing the Critical and Non-Critical Channel Alert flags as specified in [RFC7179]. The next field is the Critical Reserved bits (CRSV) that are specified herein to be the Extended Hop Count. Then the Non-Critical Reserved Bits (NCRSV) and the Critical Ingress-to-Egress bits (CITE) as specified in [RFC7179]. Finally, there is the Non-Critical Ingress-to-Egress field, the top two bits of which are specified herein as the Extended Color field.

11. IANA Considerations (Changed)

This section give IANA actions previously completed and newly requested IANA actions.

11.1 Previously Completed IANA Actions (Unchanged)

The following IANA actions were completed as part of [RFC7180] and are included here for completeness, since this document obsoletes [RFC7180].

1. The nickname 0xFFC1, which was reserved by [RFC6325], is allocated for use in the TRILL Header Egress Nickname field to indicate an OOMF (Overload Originated Multi-destination Frame).

2. Bit 1 from the seven previously reserved (RESV) bits in the per-neighbor "Neighbor RECORD" in the TRILL Neighbor TLV [RFC7176] is allocated to indicate that the RBridge sending the TRILL Hello volunteers to provide the OOMF forwarding service described in Section 2.4.2 to such frames originated by the TRILL Switch whose SNPA (MAC address) appears in that Neighbor RECORD. The description of this bit is "Offering OOMF service".

3. Bit 0 is allocated from the Capability bits in the PORT-TRILL-VER sub-TLV [RFC7176] to indicate support of the VLANs Appointed sub-TLV [RFC7176] and the VLAN inhibition setting mechanisms specified in [rfc6439bis]. The description of this bit is "Hello reduction support".

11.2 New IANA Considerations (New)

The following are new IANA actions for this document:

11.2.1 Reference Updated

All references to [RFC7180] in the TRILL Parameters Registry are replaced with references to this document except that the Reference for bit 0 in the PORT-TRILL-VER Sub-TLV Capapbilty Flags is changed to [rfc6439bis].

11.2.2 The 'E' Capability Bit

IANA is requested to allocate a previously reserved capability bit in the TRILL Version sub-TLV carried in the Router Capability and MT Capability TLVs (#242, #144) to indicate support of the [RFC7356] E-L1FS flooding scope. This capability bit is referred to as the "E" bit. The following is the addition to the

Bit	Description	References
tbd1	E-L1FS FS-LSP support	[this document][RFC7356]

11.2.3 NickFlags APPsub-TLV Number

IANA is requested to allocate an APPsub-TLV number under the TRILL GENINFO TLV from the range less than 255.

Type	Name	References
tbd2	NICKFLAGS	[this document]

11.2.4 Update TRILL Extended Header Flags

Update the "TRILL Extended Header Flags" registry as follows:

Bits	Purpose	References
14-16	Extended Hop Count	[this document]
27-28	Extended Color	[this document]
29-31	Available non-critical ingress-to-egress flags	[RFC7179] [this document]

11.2.5 TRILL-VER Sub-TLV Capability Flags

Update the "TRILL-VER Sub-TLV Capapbility Flags" registry as follows:

Bit	Description	Reference
14	Extended Hop Count support	[this document]
15-16	Unassigned	[this document]
27-28	Extended Color support	[this document]
29-31	Extended header flag support	[RFC7179] [this document]

12. Security Considerations (Changed)

This memo improves the documentation of the TRILL protocol, corrects five errata in [RFC6325], updates [RFC6325], [RFC7177], and [RFC7179] and obsoletes [RFC7180].

It does not change the Security Considerations of these RFCs to which the reader is referred. {{ Probably need to say more than this. }}

Acknowledgements

The contributions of the following individuals to this document are gratefully acknowledged:

Santosh Rajagopalan

The contributions of the following, listed in alphabetic order, to the preceding version of this document, [RFC7180], are gratefully acknowledged:

Somnath Chatterjee, Weiguo Hao, Rakesh Kumar, Yizhou Li, Radia Perlman, Mike Shand, Meral Shirazipour, and Varun Shah.

Normative References

[802.1Q-2011] - IEEE, "IEEE Standard for Local and metropolitan area networks -- Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q-2011, August 2011.

[IS-IS] - International Organization for Standardization, "Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)", Second Edition, November 2002.

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

[RFC5305] - Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5306] - Shand, M. and L. Ginsberg, "Restart Signaling for IS-IS", RFC 5306, October 2008.

[RFC6325] - Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

[RFC6361] - Carlson, J. and D. Eastlake 3rd, "PPP Transparent Interconnection of Lots of Links (TRILL) Protocol Control Protocol", RFC 6361, August 2011.

[RFC7172] - Eastlake 3rd, D., Zhang, M., Agarwal, P., Perlman, R., and D. Dutt, "Transparent Interconnection of Lots of Links (TRILL): Fine-Grained Labeling", RFC 7172, May 2014.

[RFC7176] - Eastlake 3rd, D., Senevirathne, T., Ghanwani, A., Dutt, D., and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", RFC 7176, May 2014.

[RFC7177] - Eastlake 3rd, D., Perlman, R., Ghanwani, A., Yang, H., and V. Manral, "Transparent Interconnection of Lots of Links (TRILL): Adjacency", RFC 7177, May 2014.

[RFC7179] - Eastlake 3rd, D., Ghanwani, A., Manral, V., Li, Y., and C. Bestler, "Transparent Interconnection of Lots of Links (TRILL): Header Extension", RFC 7179, May 2014, <http://www.rfc-editor.org/info/rfc7179>.

[RFC7356] - Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, September 2014, <http://www.rfc-editor.org/info/rfc7356>.

Informative References

[802] - IEEE 802, "IEEE Standard for Local and metropolitan area networks: Overview and Architecture", IEEE Std 802.1-2001, 8 March 2002.

[Err3002] - RFC Errata, Errata ID 3002, RFC 6325, <http://www.rfc-editor.org>.

[Err3003] - RFC Errata, Errata ID 3003, RFC 6325, <http://www.rfc-editor.org>.

[Err3004] - RFC Errata, Errata ID 3004, RFC 6325, <http://www.rfc-editor.org>.

[Err3052] - RFC Errata, Errata ID 3052, RFC 6325, <http://www.rfc-editor.org>.

[Err3053] - RFC Errata, Errata ID 3053, RFC 6325, <http://www.rfc-editor.org>.

[Err3508] - RFC Errata, Errata ID 3508, RFC 6325, <http://rfc-editor.org>.

[RFC4086] - Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005, <http://www.rfc-editor.org/info/rfc4086>.

[RFC6327] - Eastlake 3rd, D., Perlman, R., Ghanwani, A., Dutt, D., and V. Manral, "Routing Bridges (RBridges): Adjacency", RFC 6327, July 2011, <http://www.rfc-editor.org/info/rfc6327>.

[RFC6439] - Perlman, R., Eastlake, D., Li, Y., Banerjee, A., and F. Hu, "Routing Bridges (RBridges): Appointed Forwarders", RFC 6439, November 2011, <http://www.rfc-editor.org/info/rfc6439>.

[RFC7042] - Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 7042, October 2013.

[RFC7175] - Manral, V., Eastlake 3rd, D., Ward, D., and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL): Bidirectional Forwarding Detection (BFD) Support", RFC 7175, May 2014.

[RFC7178] - Eastlake 3rd, D., Manral, V., Li, Y., Aldrin, S., and D. Ward, "Transparent Interconnection of Lots of Links (TRILL): RBridge Channel Support", RFC 7178, May 2014.

[RFC7180] - Eastlake 3rd, D., Zhang, M., Ghanwani, A., Manral, V.,

and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL): Clarifications, Corrections, and Updates", RFC 7180, May 2014.

[RFC7357] - Zhai, H., Hu, F., Perlman, R., Eastlake 3rd, D., and O. Stokes, "Transparent Interconnection of Lots of Links (TRILL): End Station Address Distribution Information (ESADI) Protocol", RFC 7357, September 2014, <http://www.rfc-editor.org/info/rfc7357>.

[RFC7379] - Li, Y., Hao, W., Perlman, R., Hudson, J., and H. Zhai, "Problem Statement and Goals for Active-Active Connection at the Transparent Interconnection of Lots of Links (TRILL) Edge", RFC 7379, October 2014, <http://www.rfc-editor.org/info/rfc7379>.

[TRILL-OAM-FM] - Senevirathen, T., "TRILL Fault Management", draft-ietf-trill-oam-fm, work in progress.

[rfc6439bis] - Eastlake, D., et al., "TRILL: Appointed Forwarders", draft-eastlake-trill-rfc6439bis, work in progress.

Appendix A: Life Cycle of a TRILL Switch Port (New)

The contents of this informational Appendix are based on http://www.ietf.org/mail-archive/web/trill/current/msg06355.html

Question: Suppose we are developing a TRILL implementation to run on different machines. Then what happened 1st? Is LSP or ESADI started first? -> Link state database creation -> Designated RBridge election (How to set priority? any fixed process that depends on user settings?) -> etc. ?

Answer:

The first thing that happens on a port/link is any link set-up that is needed. For example, on a PPP link [RFC6361], you need to negotiate that you will be using TRILL. However, if you have Ethernet links [RFC6325], which are probably the most common type, there isn't any link set-up needed.

Then TRILL IS-IS Hellos get sent out the port to be exchanged on the link [RFC7177]. Optionally, you might also exchange MTU-probe/ack PDUs [RFC7177], BFD PDUs [RFC7175], or other link test packets. But all these other things are optional. Only Hellos are required.

TRILL doesn't send anything else on the link until the link gets out of the Down or Detect states [RFC7177].

If a link is configured as a point-to-point link, there is no Designated RBridge (DRB) election. By default, an Ethernet link is considered a LAN link and the DRB election occurs when the link is in any state other than Down. You don't have to configure priorities for each TRILL switch (RBridge) to be Designated RBridge (DRB). Things will work fine with all the RBridges on a link using default priority. But if the network manager wants to control this, they should be a way for them to configure the priorities to be DRB.

(To avoid complexity, this appendix generally describes things for a link that only has two TRILL switches on it. But TRILL works fine as currently specified on a broadcast link with multiple TRILL switches on it - actually multiple TRILL switch ports since a TRILL switch can have multiple ports connected to the same link. The most likely way to get such a multi-access link with current technology is to have more than 2 TRILL switch Ethernet ports connected to a bridged LAN. Since the TRILL protocol operates above all bridging, to the first approximation, the bridge LAN looks like a transparent broadcast link to TRILL.)

When a link gets to the 2-Way or Report state, then LSP, CSNP, and PSNP start to flow on the link (as well as FS-LSPs, FS-CSNPs, and

FS-PSNPs if the TRILL switch is using E-L1FS (see Section 8.1)).

When a link gets to the Report state, then there is adjacency. The existence of that adjacency is flooded (reported) to the campus in LSPs. TRILL data packets can then start to flow on the link as TRILL switches recalculate the least cost paths and distribution trees to take the new adjacency into account. (Until it gets to the Report state, there is no adjacency and no TRILL data packets can flow over that link (with the minor corner case exception that an RBridge Channel message can, for its first hop only, be sent on a port where there is no adjacency (Section 2.4 of [RFC7178]).) (Although this paragraph seems to be talking about link state, it is actually port state. It is possible for different TRILL switch ports on the same link to temporarily be in different states. The adjacency state machinery runs independently on each port.)

ESADI [RFC7357] is built on top of the regular TRILL routing. Since ESADI PDUs look, to transit TRILL switches, like regular TRILL data packets, no ESADI PDUs can flow until adjacencies are established and TRILL data is flowing. Of course, ESADI is optional and is not used unless configured...

Question: Does it require TRILL Full headers at the time TRILL-LSPs start being broadcast on a link? Because at that time it’s not defined Egress and Ingress nicknames.

Answer:
TRILL Headers are only for TRILL Data packets. TRILL IS-IS packets, such as TRILL-LSPs, are sent in a different way that does not use a TRILL Header and does not depend on nicknames.

Probably, in most implementations, a TRILL switch will start up using the same nickname it had when it shut down or last got disconnected from a campus. If you want, you can implement TRILL to come up initially not reporting any nickname (by not including an Nickname sub-TLV in its LSPs) until you get the link state database or most of the link state database, and then choose a nickname no other TRILL switch in the campus is using. Of course, if a TRILL switch does not have a nickname, then it cannot ingress data, cannot egress known unicast data, and cannot be a tree root.

TRILL IS-IS and LSPs and the link state database all work based on the 7-byte IS-IS System-ID (sometimes called the LAN ID). System-IDs always have to be unique across the campus so there is no problem determining topology regardless of nickname state. The Nickname system is built on top of that.

Appendix B: Example TRILL PDUs (New)

[Three for four example PDUs to be included here to help answer any questions about bit ordering or the like.]

Appendix C: Appointed Forwarder Status Lost Couter (New)

This appendix is derived from http://www.ietf.org/mail-archive/web/trill/current/msg05279.html.

Strict conformance to the provisions of Section 4.8.3 of [RFC6325] on the value of the Appointed Forwarder Status Lost Counter can result in splitting of Interested VLANs and Spanning Tree Roots sub-TLVs [RFC7176], or the corresponding Interested Lables sub-TLVs, due to minor/accidental differences in the counter value for different VLANs or FGLs.

This counter is a mechanism to optimize data plane learning by trimming the expiration timer for learned addresses on a per VLAN/FGL basis under some circumstances. Note the following:

(1) If an implementer don’t care about that optimization and don’t mind some time outs being longer than they otherwise would be, you can just not bother changing the counter, even if you are using data plane learning. On the other hand, if you don’t care about sone time outs being shortened when they otherwise wouldn’t, you could increment the counter for multiple VLANs even you don’t lose AF status on a port for all those VLANS but, for example, only one of them.

(2) If you are relying on ESADI [RFC7357] or Directory Assist [RFC7379] and not learning from the data plane, the counter doesn’t matter and there really isn’t any need to increment it.

(3) If an RBridge port has been configured with the "disable end station traffic" bit on (also known as the trunk bit), then it makes no difference if that port is appointed forwarder or not even though, according to the standard, the Appointed Forwarder selection mechanism continues to operate. So, under such circumstances, there is no reason to increment the counter if such a port loses Appointed Forwarder status.

(4) If you are updating the counter, incrementing it by more than one (even up to incrementing it by a couple of hundred), so that it matches the counter for some adjacent VLAN for the same RBridge would have an extremely small probability of causing any sub-optimization and, if it did, that sub-optimization would just be to occasionally fail to specially decrease the time out for some learned addresses.

Appendix D: Changes from [RFC7180]

This informational Appendix summarizes the changes, augmentations, and excisions this document makes to [RFC7180].

D.1 Changes

For each heading in this document ending with "(Changed)", this section summarizes how it was changed:

Section 1, Introduction: numerous changes to reflect the overall changes in contents.

Section 1.1, Precedence: changed to add mention of [RFC7179].

Section 1.3, Terminology and Acronyms: numerous terms added.

Section 3, Distribution Trees and RPF Check: changed by the addition of the new material in Section 3.6. See C.2 item 1.

Section 8, Other IS-IS Considerations: Changed by the addition of Sections 8.1, 8.2, 8.3, and 8.4. See Appendix C.2 items 2, 3, 4, and 5 respectively.

Section 9, Updates to [RFC7177] (Adjacency): Changes and additions to [RFC7177] to support E-L1FS. See Appendix C.2, item 2.

Section 11, IANA Considerations: changed by the addition of material in Section 11.2. See Appendix C.2, item 7.

Section 12, Security Considerations: minor changes in the RFCs listed.

D.2 Additions

The following material was added to [RFC7180] in producing this document:

1. Addition of support for an alternative Reverse Path Forwarding Check (RPFC) along with considerations for deciding between the original [RFC6325] RPFC and this alternative RPFC. This alternative RPFC was originally discussed on the TRILL WG mailing list in http://www.ietf.org/mail-archive/web/trill/current/msg01852.html and subsequent messages. (Section 3.6)

2. Addition of mandatory E-L1FS [RFC7356] support (Section 8.1, Section 9).

3. Recommendations concerning control packet priorities. (Section 8.2)

4. Implementation requirements concerning unknown IS-IS PDU types (Section 8.3).

5. Specification of an optional Nickname Flags APPsub-TLV and an ingress flag within that APPsub-TLV. (Section 8.4)

6. Update TRILL Header to allocate a Color bit (Section 10.1) and update the optional TRILL Header Extension Flag Word to allocate a two-bit Extended Color field (Section 10.2).

7. Some new IANA Considerations in Section 11.2.

8. Informative Appendix A and C on the Lifecycle of a TRILL Port and the Appointed Forwarder Status Lost Counter, respectively.

9. Appendix B with example TRILL PDUs.

D.3 Deletions

The following material was deleted from [RFC7180] in producing this document:

1. Removal of all updates to [RFC6327] that occurred in [RFC7180]. These have been rolled into [RFC7177] that obsoletes [RFC6327]. However, new updates to [RFC7177] are included (see Item 1 in Section A.1).

2. Removal of all updates to [RFC6439]. These have been rolled into [rfc6439bis] that will obsolete [RFC6439].

Appendix Z: Change History

This appendix lists version changes in this document.

From -00 to -01

1. Update Author Addresses.
2. Add Appendix C moving previous Appendix C to D.
3. Change the upper four bits of the former Ex-Length field in the TRILL Header to be reserved.
4. Minor editorial changes.

Authors' Addresses

Donald Eastlake 3rd
Huawei Technology
155 Beaver Street
Milford, MA 01757 USA

Phone: +1-508-333-2270
EMail: d3e3e3@gmail.com

Mingui Zhang
Huawei Technologies
No. 156 Beiqing Rd. Haidian District,
Beijing 100095
P.R. China

EMail: zhangmingui@huawei.com

Radia Perlman
EMC
2010 256th Avenue NE, #200
Bellevue, WA 98007 USA

Email: radia@alum.mit.edu

Ayan Banerjee
Cisco

EMail: ayabaner@cisco.com

Anoop Ghanwani
Dell
5450 Great America Parkway
Santa Clara, CA 95054 USA

EMail: anoop@alumni.duke.edu

Sujay Gupta
IP Infusion,
RMZ Centennial
Mahadevapura Post
Bangalore - 560048 India

EMail: sujay.gupta@ipinfusion.com

Copyright and IPR Provisions

TRILL

Weiguo Hao
Yizhou Li
Tao Han
Huawei
S. Hares
Hickory Hill Consulting
Muhammad Durrani
Brocade
Sujay Gupta
IP Infusion

Internet Draft

Intended status: Standard Track

August 29, 2014

Expires: February 2015

Centralized Replication for BUM traffic in active-active edge connection

draft-hao-trill-centralized-replication-02.txt

Status of this Memo

This Internet-Draft will expire on November 29, 2014.

Copyright Notice

Abstract

In TRILL active-active access scenario, RPF check failure issue may occur when pseudo-nickname mechanism in [TRILLPN] is used. This draft describes a solution to the RPF check failure issue through centralized replication for BUM (Broadcast, Unknown unicast, Mutlicast) traffic. The solution has all ingress RBs send BUM

traffic to a centralized node via unicast TRILL encapsulation. When the centralized node receives the BUM traffic, it decapsulates the traffic and forwards the BUM traffic to all destination RBs using a distribution tree established via the TRILL base protocol. To avoid RPF check failure on a RBridge sitting between the ingress RBridge and the centralized replication node, some change of RPF calculation algorithm is required. RPF calculation on each RBridge should use the centralized node as ingress RB instead of the real ingress RBridge of RBv to perform the calculation.

Table of Contents

1. Introduction

The IETF TRILL (Transparent Interconnection of Lots of Links) [RFC6325] protocol provides loop free and per hop based multipath data forwarding with minimum configuration. TRILL uses IS-IS [RFC6165] [RFC6326bis] as its control plane routing protocol and defines a TRILL specific header for user data.

Classic Ethernet device (CE) devices typically are multi-homed to multiple edge RBridges which form an edge group. All of the uplinks of CE are bundled as a Multi-Chassis Link Aggregation (MC-LAG). An active-active flow-based load sharing mechanism is normally implemented to achieve better load balancing and high reliability. A CE device can be a layer 3 end system by itself or a bridge switch through which layer 3 end systems access to TRILL campus.

In active-active access scenario, pseudo-nickname solution in [TRILLPN] can be used to avoid MAC flip-flop on remote RBs. The basic idea is to use a virtual RBridge of RBv with a single pseudo-nickname to represent an edge group that MC-LAG connects to. Any member RBridge of that edge group should use this pseudo-nickname rather than its own nickname as ingress nickname when it injects TRILL data frames to TRILL campus. The use of the nickname solves the address flip flop issue by making the MAC address learnt by the remote RBridge bound to pseudo-nickname. However, it introduces another issue, which is incorrect packet drop by RPF check failure. Due to edge RBridges which use a pseudo-nickname other than own nicknames as the ingress nickname (Eg. Nick-Y) when the RBbridge forwards BUM traffic from local CE, the traffic will be treated by an RBridge (RBn) sitting between the ingress RB and distribution tree root as traffic whose ingress point is the virtual RBridge of RBv. If same distribution tree is used by these different edge RBridges, the traffic may arrive at RBn from different ports. Then the RPF check fails, and some of the traffic receiving from unexpected ports will be dropped by RBn.

This document proposes a centralized replication solution for broadcast, unknown unicast, multicast(BUM) traffic to solve the issue of incorrect packet drop by RPF check failure. The basic idea is that all ingress RBs send BUM traffic to a centralized node which is recommended to be a distribution tree root using unicast TRILL encapsulation. When the centralized node receives that traffic, it decapsulates it and then forwards the BUM traffic to all destination RBs using a distribution tree established as per TRILL base protocol.

2. Conventions used in this document

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].The acronyms and terminology in [RFC6325] is used herein with the following additions:

BUM - Broadcast, Unknown unicast, and Multicast

CE - As in [CMT], Classic Ethernet device (end station or bridge).

The device can be either physical or virtual equipment.

3. Centralized Replication Solution Overview

When an edge RB receives BUM traffic from a CE device, it acts as ingress RB and uses unicast TRILL encapsulation instead of multicast TRILL encapsulation to send the traffic to a centralized node. The centralized node is recommended to be a distribution tree root.

The TRILL header of the unicast TRILL encapsulation contains an "ingress RBridge nickname" field and an "egress RBridge nickname" field. If ingress RB receives the traffic from the port which is in a MC-LAG, it should set the ingress RBridge nickname to be the pseudo-nickname to avoid MAC flip-flop on remote RBs as per [TRILLPN]. Otherwise the ingress nickname should be set to ingress RBridge’s own nickname. The egress RBridge nickname is set to the special nickname of the centralized node which is used to differentiate the unicast TRILL encapsulation BUM traffic from normal unicast TRILL traffic.

When the centralized node receives the unicast TRILL encapsulated BUM traffic from ingress RB, the node decapsulates the packet. Then the centralized node replicates and forwards the BUM traffic to all destination RBs using one of the distribution trees established as per TRILL base protocol, if the centralized node is the root of a distribution tree, the recommended distribution tree is the tree whose root is the centralized node itself. When the centralized node forwards the BUM traffic, ingress nickname remains the same as that in frame it received to ensure that the MAC address learnt by all egress RBridges bound to pseudo-nickname.

When the replicatedtraffic is forwarded on each RBridge along the distribution tree starting from the centralized node, RPF check will be performed as per RFC6325. For any RBridge sitting between the ingress RBridge and the centralized replication node, the traffic incoming port should be the centralized node facing port as the multicast traffic always comes from the centralized node in this solution. However the RPF port as result of distribution tree calculation as per RFC 6325 will be the real ingress RB facing port as it uses virtual RBridge as ingress RB, so RPF check will fail. To solve this problem, some change of RPF calculation algorithm is required. RPF calculation on each RBridge should use the centralized node as ingress RB instead of the real ingress virtual RBridge to perform the calculation. As a result, RPF check will point to the centralized node facing port on the RBridge for multi-destination traffic. It prevents the incorrect frame discard by RPF check.

To differentiate the unicast TRILL encapsulation BUM traffic from normal unicast TRILL traffic on a centralized node, besides the

centralized node's own nickname, a special nickname should be introduced for centralized replication. Only when the centralized node receives unicast TRILL encapsulation traffic with egress nickname equivalent to the special nickname, the node does unicast TRILL decapsulaton and then forwards the traffic to all destination RBs through a distribution tree. The centralized nodes should announce its special use nickname to all TRILL campus through TRILL LSP extension.

4. Frame duplication from remote RB

Frame duplication may occur when a remote host sends multi-destination frame to a local CE which has an active-active connection to the TRILL campus. To avoid local CE receiving multiple copies from a remote RBridge, the designated forwarder (DF) mechanism should be supported for egress direction multicast traffic.

DF election mechanism allows only one port in one RB of MC-LAG to forward multicast traffic from TRILL campus to local access side for each VLAN. The basic idea of DF is to elect one RBridge per VLAN from an edge group to be responsible for egressing the multicast traffic. [draft-hao-trill-dup-avoidance-active-active-02] describes the detail DF mechanism and TRILL protocol extension for DF election.

If DF-election mechanism is used for frame duplication prevention, access ports on an RB are categorized as three types: non mc-lag, mc-lag DF port and mc-lag non-DF port. The last two types can be called mc-lag port. For each of the mc-lag port, there is a pseudo-nickname associated. If consistent nickname allocation per edge group RBridges is used, it is possible that same pseudo-nickname associated to more than one port on a single RB. A typical scenario is that CE1 is connected to RB1 & RB2 by mc-lag1 while CE2 is connected to RB1 & RB2 by mc-lag 2. In order to save the number of pseudo-nickname used, member ports for both mc-lag1 and mc-lag2 on RB1 & RB2 are all associated to pseudo-nickname pn1.

5. Local forwarding behavior on ingress RBridge

When a ingress RBridge(RB1) receives BUM traffic from an active-active accessing CE(CE1) device, the traffic will be injected to TRILL campus through TRILL encapsulation, and it will be replicated and forwarded to all destination RBs which include ingress RB itself along a TRILL distribution tree. So the traffic will return to the ingress RBridge. To avoid the traffic looping back to original sender CE, ingress nickname can be used for traffic filtering.

If there are two local connecting CE(CE1 and CE2) devices on ingress RB, the BUM traffic between these two CEs can't be forwarded locally and through TRILL campus simultaneously, otherwise duplicated traffic will be received by destination CE. Local forwarding behavior on ingress RBridge should be carefully designed.

To avoid duplicated traffic on receiver CE, local replication behavior on RB1 is as follows:

1. Local replication to the ports associated with the same pseudo-nickname as that associated to the incoming port as per RFC6325.

2. Do not replicate to mc-lag port associated with different pseudo-nickname.

3. Do not replicate to non mc-lag ports.

The above local forwarding behavior on the ingress RB of RB1 can be called centralized local forwarding behavior A.

If ingress RB of RB1 itself is the centralized node, BUM traffic injected to TRILL campus won't loop back to RB1. In this case, the local forwarding behavior is called centralized local forwarding behavior B. The local replication behavior on RB1 is as follows:

1. Local replication to non mc-lag ports as per RFC6325.

2. Local replication to the ports associated with the same pseudo-nickname as that associated to the incoming port as per RFC6325.

3. Local replication to the mc-lag DF port associated with different pseudo-nickname as per RFC6325. Do not replicate to mc-lag non-DF port associated with different pseudo-nickname.

6. Loop prevention among RBridges in a edge group

If a CE sends a broadcast, unknown unicast, or multicast (BUM) packet through DF port to a ingress RB, it will forward that packet to all or subset of the other RBs that only have non-DF ports for that MC-LAG. Because BUM traffic forwarding to non-DF port isn't allowed, in this case the frame won't loop back to the CE.

If a CE sends a BUM packet through non-DF port to a ingress RB, say RB1, then RB1 will forward that packet to other RBridges that have DF port for that MC-LAG. In this case the frame will loop back to the CE and traffic split-horizon filtering mechanism should be used to avoid looping back among RBridges in a edge group.

Split-horizon mechanism relies on ingress nickname to check if a packet's egress port belongs to a same MC-LAG with the packet's incoming port to TRILL campus.

When the ingress RBridge receives BUM traffic from an active-active accessing CE device, the traffic will be injected to TRILL campus through TRILL encapsulation, and it will be replicated and forwarded to all destination RBs which include ingress RB itself through TRILL distribution tree. If same pseudo-nickname is used for two active-active access CEs as ingress nickname, egress RB can use the nickname to filter traffic forwarding to all local CE. In this case, the traffic between these two CEs goes through local RB and another copy of the traffic from TRILL campus is filtered. If different ingress nickname is used for two connecting CE devices, the access ports connecting to these two CEs should be isolated with each other. The BUM traffic between these two CEs should go through TRILL campus, otherwise the destination CE connected to same RB with the sender CE will receive two copies of the traffic.

Do note that the above sections on techniques to avoid frame duplication, loop prevention is applicable assuming the Link aggregation technology in use is unaware of the frame duplication happening. For example using mechanisms like IEEE802.1AX, Distributed Resilient Network Interconnect (DRNI) specs implements mechanism similar to DF and also avoids some cases of frame duplication & looping.

7. Centralized replication forwarding process

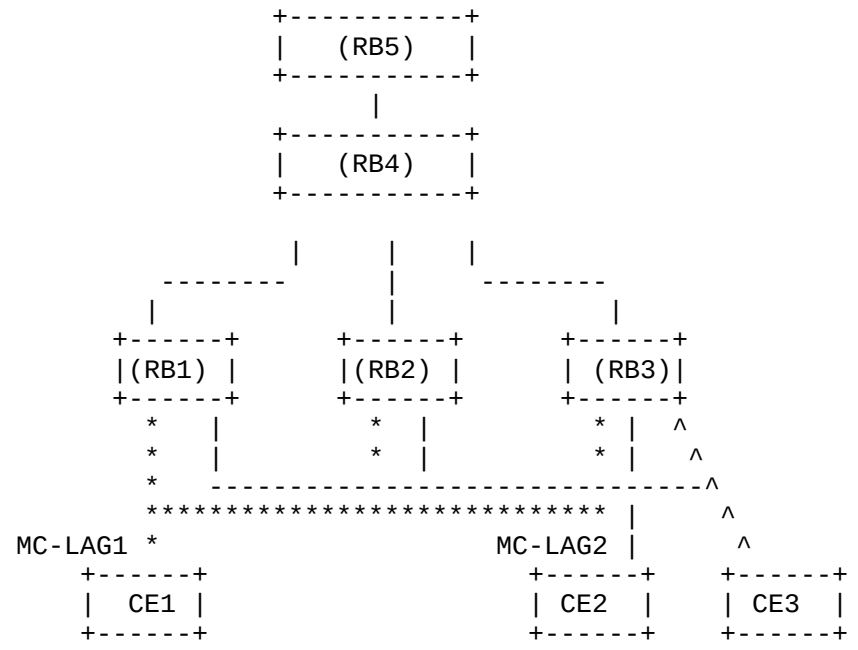


Figure 1 TRILL Active-active access

Assuming the centralized replication solution is used in the network of above figure 1, RB5 is the distribution tree root and centralized replication node, CE1 and CE2 are active-active accessed to RB1,RB2 and RB3 through MC-LAG1 and MC-LAG2 respectively, CE3 is single homed to RB3. The RBridge's own nickname of RB1 to RB5 are nick1 to nick5 respectively. RB1,RB2 and RB3 use same pseudo-nickname for MC-LAG1 and MC-LAG2, the pseudo-nickname is P-nick. The special use nickname on the centralized replication node of RB5 is S-nick.

The BUM traffic forwarding process from CE1 to CE2,CE3 is as follows:

1. CE1 sends BUM traffic to RB3.

2. RB3 replicates and sends the BUM traffic to CE2 locally. RB2 also sends the traffic to RB5 through unicast TRILL encapsulation. Ingress nickname is set as P-nick, egress nickname is set as S-nick.

3. RB5 decapsulates the unicast TRILL packet. Then it uses the distribution tree whose root is RB5 to forward the packet. The egress nickname in the trill header is the nick5. Ingress nickname is still P-nick.

4. RB4 receives multicast TRILL traffic from RB5. Traffic incoming port is the up port facing to distribution tree root, RPF check will be correct based on the changed RPF port calculation algorithm in this document. After RPF check is performed, it forwards the traffic to all other egress RBs(RB1,RB2 and RB3).

5. RB3 receives multicast TRILL traffic from RB4. It decapsulates the multicast TRILL packet. Because ingress nickname of P-nick is equivalent to the nickname of local MC-LAGs connecting CE1 and CE2, it doesn’t forward the traffic to CE1 and CE2 to avoid duplicated frame. RB3 only forwards the packet to CE3.

6. RB1 and RB2 receive multicast TRILL traffic from RB4. The forwarding process is similar to the process on RB3, i.e, because ingress nickname of P-nick is equivalent to the nickname of local MC-LAGs connecting CE1 and CE2, they also don’t forward the traffic to local CE1 and CE2.

8. BUM traffic loadbalancing among multiple centralized nodes

To support unicast TRILL encapsulation BUM traffic load balancing, multiple centralized replication node can be deployed and the traffic can be load balanced on these nodes in vlan-based or flow-based mode.

8.1. Vlan-based loadbalancing

Assuming there are k centralized nodes in TRILL campus, VLAN-based(or FGL-based, etc) loadbalancing algorithm used by ingress active-active access RBridge is as follows:

1. All centralized nodes are ordered and numbered from 0 to k-1

in ascending order according to the 7-octet IS-IS ID.

2. For VLAN ID m, choose the centralized node whose number equals (m mod k).

An example of the m mod K, is that for 3 centralized nodes (CN) and 5 VLANs is: VLAN 0 goes to CN0, VLAN1 goes to CN1, VLAN2 goes to CN2, VLAN4 goes to CN0, and VLAN5 goes to CN1.

When a ingress RBridge participating active-active connection receives BUM traffic from local CE, the RB decides to send the traffic to which centralized node based on the VLAN-based loadbalancing algorithm, vlan-based loadbalancing for the BUM traffic can be achieved among multiple centralized nodes.

8.2. Flow-based loadbalancing

To support flow-based loadbalancing for BUM traffic between different centralized node, anycast special use nickname mechanism should be introduced, which means a same special use nickname is attached to both physical centralized node at the same time. Each centralized node announces the special use nickname through the Nickname Sub-Tlv specified in [RFC6326] to TRILL network and MUST ignore the nickname collision check as defined in basic TRILL protocol.

The egress nickname of unicast TRILL encapsulation for BUM traffic from ingress RB is the special use nickname. The unicast TRILL encapsulation BUM traffic would go to any one of the physical centralized nodes by the natural support of ECMP from TRILL protocol.

The physical centralized node will decapsulate the unicast TRILL encapsulation and forwards it through any one of the distribution trees established per RFC 6325 with the original source, and BUM destination. Because ECMP of the unicast TRILL encapsulation BUM traffic is supported among multiple centralized nodes, so it can achieve better link bandwidth usage than VLAN-based(or FGL-based, etc)loadbalancing.

9. Network Migration Analysis

Centralized nodes need software and hardware upgrade to support centralized replication process, which stitches TRILL unicast traffic decapsulation process and the process of normal TRILL multicast traffic forwarding along distribution tree.

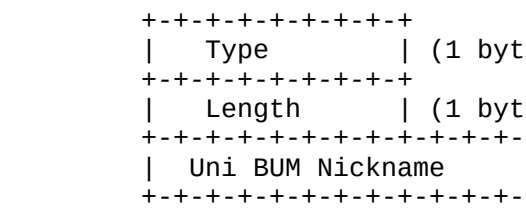
Active-active connection edge RBs need software and hardware upgrade to support unicast TRILL encapsulation for BUM traffic, the process is similar to normal head-end replication process.

Transit nodes need software upgrade to support RPF port calculation algorithm change.

10. TRILL protocol extension

The Unicast BUM Nickname TLV is introduced to announce its special use nickname for centralized replication by centralized node. It is carried in an LSP PDU. Ingress RBs rely on the TLV to learn the egress nickname of TRILL unicast encapsulation for BUM traffic.

10.1. The Unicast BUM Nickname sub-TLV



o Type: Router Capability sub-TLV type, TBD (Uni-BUM-VLANs).

o Length: indicates the length of Uni BUM Nickname field, it is a fixed value of 4.

o Uni BUM Nickname: The nickname is exclusively used for centralized replication solution purpose. Ingress RBs use the nickname as egress nickname in trill header of unicast TRILL encapsulation for BUM traffic.

11. Security Considerations

This draft does not introduce any extra security risks. For general TRILL Security Considerations, see [RFC6325].

12. IANA Considerations

This document requires no IANA Actions. RFC Editor: Please remove this section before publication.

13. References

13.1. Normative References

[1] [RFC6165] Banerjee, A. and D. Ward, "Extensions to IS-IS for Layer-2 Systems", RFC 6165, April 2011.

[2] [RFC6325] Perlman, R., et.al. "RBridge: Base Protocol Specification", RFC 6325, July 2011.

[3] [RFC6326bis] Eastlake, D., Banerjee, A., Dutt, D., Perlman, R., and A. Ghanwani, "TRILL Use of IS-IS", draft-eastlake-isis-rfc6326bis, work in progress.

13.2. Informative References

[4] [TRILLPN] Zhai,H., et.al., "RBridge: Pseduonode nickname", draft-hu-trill-pseudonode-nickname, Work in progress, November 2011.

[5] [TRILAA] Li,Y., et.al., " Problem Statement and Goals for Active-Active TRILL Edge", draft-ietf-trill-active-active-connection-prob-00, Work in progress, July 2013.

[6] [CMT] Senevirathne, T., Pathangi, J., and J. Hudson, "Coordinated Multicast Trees (CMT)for TRILL", draft-ietf-trill-cmt-00.txt Work in Progress, April 2012.

14. Acknowledgments

The authors wish to acknowledge the important contributions of Hongjun Zhai, Xiaomin Wu, Liang Xia.

Authors' Addresses

Weiguo Hao
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China
Phone: +86-25-56623144
Email: haoweiguo@huawei.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China
Phone: +86-25-56625375
Email: liyizhou@huawei.com

Tao Han
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China
Phone: +86-25-56623454
Email: billow.han@huawei.com

Susan Hares
Hickory Hill Consulting
7453 Hickory Hill
Saline, CA 48176
USA
Email: shares@ndzh.com

Muhammad Durrani
Brocade communications Systems, Inc
EMail: mdurrani@Brocade.com

Sujay Gupta
IP Infusion,
RMZ Centennial
Mahadevapura Post
Bangalore - 560048
India
EMail: sujay.gupta@ipinfusion.com

INTERNET-DRAFT
Intended Status: Proposed Standard

Mingui Zhang
Huawei
Radia Perlman
EMC
Hongjun Zhai
JIT
Muhammad Durrani
Brocade
Sujay Gupta
IP Infusion

Expires: February 7, 2016
August 6, 2015

TRILL Active-Active Edge Using Multiple MAC Attachments
draft-ietf-trill-aa-multi-attach-04.txt

Abstract

TRILL active-active service provides end stations with flow level load balance and resilience against link failures at the edge of TRILL campuses as described in RFC 7379.

This draft specifies a method by which member RBridges in an active-active edge RBridge group use their own nicknames as ingress RBridge nicknames to encapsulate frames from attached end systems. Thus, remote edge RBridges are required to keep multiple locations of one MAC address in one Data Label. Design goals of this specification are discussed in the document.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Table of Contents

1. Introduction

As discussed in [RFC7379], in a TRILL Active-Active Edge (AAE) topology, a Local Active-Active Link Protocol (LAALP), for example, a Multi-Chassis Link Aggregation Group (MC-LAG), is used to connect multiple RBridges to multi-port Customer Equipment (CE), such as a switch, vSwitch or a multi-port end station. A set of endnodes are attached in the case of switch or vSwitch. It is required that data traffic within a specific VLAN from this endnode set (including the multi-port end station case) can be ingressed and egressed by any of these RBridges simultaneously. End systems in the set can spread their traffic among these edge RBridges at the flow level. When a link fails, end systems keep using the remaining links in the LAALP without waiting for the convergence of TRILL, which provides resilience to link failures.

Since a frame from each endnode can be ingressed by any RBridge in the local AAE group, a remote edge RBridge may observe multiple attachment points (i.e., egress RBridges) for this endnode. This issue is known as the "MAC flip-flopping".

In this document, AAE member RBridges use their own nicknames to ingress frames into the TRILL campus. Remote edge RBridges are required to keep multiple points of attachment per MAC address and Data Label attached to the AAE. This addresses the MAC flip-flopping issue. The use of the solution, as specified in this document, in an AAE group does not prohibit the use of other solutions in other AAE groups in the same TRILL campus. For example, the specification in this draft and the specification in [PN] could be simultaneously deployed for different AAE groups in the same campus.

The main body of this document is organized as follows. Section 2 lists acronyms and terminologies. Section 3 gives the overview model. Section 4 provides options for incremental deployment. Section 5 describes how this approach meets the design goals. The Sections after Section 5 cover security, IANA, and some backwards compatibility considerations.

2. Acronyms and Terminology

2.1. Acronyms and Terms

AAE: Active-Active Edge

Campus: a TRILL network consisting of TRILL switches, links, and possibly bridges bounded by end stations and IP routers. For TRILL, there is no "academic" implication in the name "campus".

CE: Customer Equipment (end station or bridge). The device can be either physical or virtual equipment.

Data Label: VLAN or FGL

DRNI: Distributed Resilient Network Interconnect. A link aggregation specified in [802.1AX] that can provide an LAALP between from 1 to 3 CEs and 2 or 3 RBridges.

Edge RBridge: An RBridge providing end station service on one or more of its ports.

E-L1FS: Extended Level 1 Flooding Scope

ESADI: End Station Address Distribution Information [RFC7357]

FGL: Fine Grained Label [RFC7172]

FS-LSP: Flooding Scoped Link State PDU

IS: Intermediate System [ISIS]

IS-IS: Intermediate System to Intermediate System [ISIS]

LAALP: As in [RFC7379], Local Active-Active Link Protocol. Any protocol similar to MC-LAG (or DRNI) that runs in a distributed fashions on a CE, the links from that CE to a set of edge group RBridges, and on those RBridges.

LSP: Link State PDU

MC-LAG: Multi-Chassis LAG. Proprietary extensions of Link Aggregation [802.1AX] that can provide an LAALP between one CE and 2 or more RBridges.

PDU: Protocol Data Unit

RBridge: A device implementing the TRILL protocol.

TRILL: TRansparent Interconnection of Lots of Links or Tunneled Routing in the Link Layer [RFC6325] [RFC7177].

TRILL switch: An alternative name for an RBridge.

vSwitch: A virtual switch such as a hypervisor that also simulates a bridge.

2.2. Terminology

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].

Familiarity with [RFC6325], [RFC6439] and [RFC7177] is assumed in this document.

3. Overview

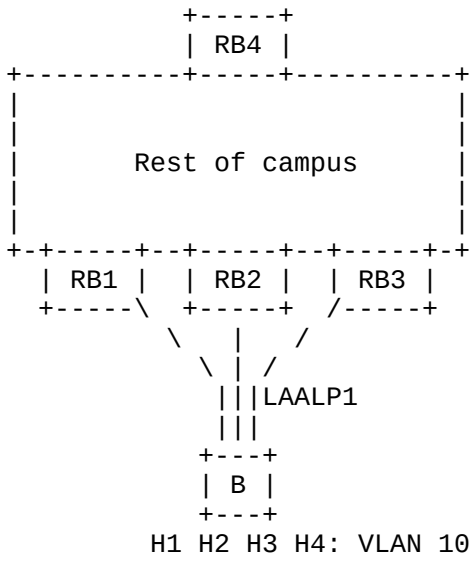


Figure 3.1: An example topology for TRILL Active-Active Edge

Figure 3.1 shows an example network for TRILL Active-Active Edge. In this figure, endnodes (H1, H2, H3 and H4) are attached to a bridge B that communicates with multiple RBridges (RB1, RB2 and RB3) via the LAALP. Suppose RB4 is a ’remote’ RBridge not in the AAE group in the TRILL campus. This connection model is also applicable to the virtualized environment where the physical bridge can be replaced with a vSwitch while those bare metal hosts are replaced with virtual machines (VM).

For a frame received from its attached endnode sets, a member RBridge of the AAE group conforming to this document always encapsulates that frame using its own nickname as the ingress nickname no matter whether it’s unicast or multicast.

With the options specified as follows, even though the remote RBridge RB4 will see multiple attachments for each MAC from one of the end-nodes, the "MAC flip-flopping" will not cause any problem.

4. Incremental Deployable Options

Two options are specified. Option A requires new hardware support. Option B can be incrementally implemented throughout a TRILL campus with common existing TRILL fast path hardware. Further details on Option B are given in Section 4.1.

-- Option A

A new capability announcement would appear in LSPs: "I can cope with data plane learning of multiple attachments for an endnode". This mode of operation is generally not supported by existing TRILL fast path hardware. Only if all edge RBridges, to which the group has data connectivity, and that are interested in any of the Data Labels in which the AAE is interested, announce this capability, can the AAE group safely use this approach. If all such RBridges do not announce this "Option A" capability, then a fallback would be needed such as reverting from active-active to active-standby operation or isolating the RBridges that would need to support this capability and do not support it. Further details for Options A are beyond the scope of this document except that in Section 4.2 a bit is reserved to indicate support for Option A because a remote RBridge supporting Option A is compatible with an AAE group using Option B.

-- Option B

As pointed out in Section 4.2.6 of [RFC6325] and Section 5.3 of [RFC7357], one MAC address may be persistently claimed to be attached to multiple RBridges within the same Data Label in the TRILL ESADI-LSPs. For Option B, AAE member RBridges make use of the TRILL ESADI protocol to distribute multiple attachments of a MAC address. Remote RBridges SHOULD disable the data plane MAC learning for such multi-attached MAC addresses from TRILL Data packet decapsulation unless they also support Option A. The ability to configure an RBridge to disable data plane learning is provided by the base TRILL protocol [RFC6325].

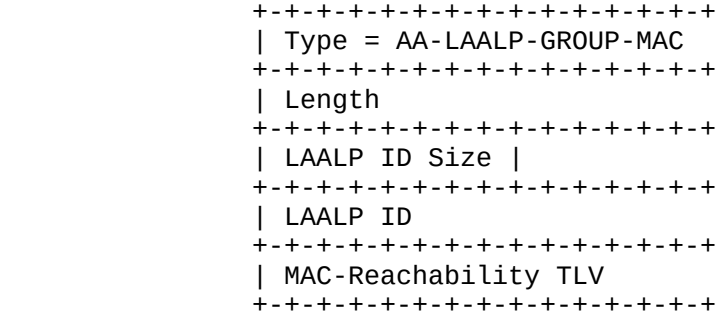
4.1. Details of Option B

With Option B, an RBridge in an AAE group MUST advertise all Data Labels enabled for all its attached LAALPs and participate in ESADI for those Data Labels. The receiving edge RBridges MUST avoid flip-flop errors in MAC learned from the TRILL Data packet decapsulation for the originating RBridge within these Data Labels. It’s RECOMMENDED that the receiving edge RBridge disable the data plane MAC learning from TRILL Data packet decapsulation within those advertised Data Labels for the originating RBridge unless the

receiving RBridge also supports Option A. However, alternative implementations MAY be used to produce the same expected behavior. A promising way is to make use of the confidence level mechanism [RFC6325]. For example, let the receiving edge RBridge give a prevailing confidence value (e.g., 0x21) to the first MAC attachment learned from the data plane over others from the TRILL Data packet decapsulation. So the receiving edge RBridge will stick to this MAC attachment until it is overridden by one learned from the ESADI protocol [RFC7357]. The MAC attachment learned from ESADI is set to have higher confidence value (e.g., 0x80) to override any alternative learning from the decapsulation of received TRILL Data packets [RFC6325].

Enabled Data Labels for an LAALP are advertised by allocating one reserved flag from the Interested VLANs and Spanning Tree Roots Sub-TLV (Section 2.3.6 of [RFC7176]) and one reserved flag from the Interested Labels and Spanning Tree Roots Sub-TLV (Section 2.3.8 of [RFC7176]). When this flag is set to 1, the originating IS (RBridge) is advertising Data Labels for LAALPs rather than plain LAN links. (See Section 8.3)

Whenever a MAC from the LAALP of this AAE is learned through ingress or configuration, it MUST be advertised via the ESADI protocol [RFC7357]. In its TRILL ESADI-LSPs, the originating RBridge needs to include the identifier of this AAE. Remote RBridges need to know all nicknames of RBridges in this AAE. This is achieved by listening to the "AA LAALP Group RBridges" TRILL APPsub-TLV defined in Section 5.3.2. The MAC Reachability TLVs [RFC6165] are composed in a way that each TLV only contains MAC addresses of end-nodes attached to a single LAALP. Each such TLV is enclosed in a TRILL APPsub-TLV defined as follows.



- Type: AA LAALP Grouped MAC (TRILL APPsub-TLV type tbd1)

- Length: The MAC-Reachability TLV [RFC6165] is contained in the value field as a sub-TLV. The total number of bytes contained in

the value field is given by k+8+6*n.

- LAALP ID Size: The length k of the LAALP ID in bytes.

- LAALP ID: The ID of the LAALP that is k bytes long. Here, it also serves as the identifier of the AAE. If the LAALP is an MC-LAG (or DRNI), it is the 8 byte ID as specified in Clause 6.3.2 in [802.1AX].

- MAC-Reachability sub-TLV: The AA-LAALP-GROUP-MAC APPsub-TLV value contains the MAC-Reachability TLV as a sub-TLV. As specified in Section 2.2 in [RFC7356], the type and length fields of the MAC-Reachability TLV are encoded as unsigned 16 bit integers. The one octet unsigned Confidence along with these TLVs SHOULD be set to prevail over those MAC addresses learned from TRILL Data decapsulation by remote edge RBridges.

This AA-LAALP-GROUP-MAC APPsub-TLV MUST be included in a TRILL GENINFO TLV [RFC7357] in the ESADI-LSP. There may be more than one occurrence of such TRILL APPsub-TLV in one ESADI-LSP fragment.

For those MAC addresses contained in an AA-LAALP-GROUP-MAC APPsub-TLV, this document applies. Otherwise, [RFC7357] applies. For example, an AAE member RBridge continues to enclose MAC addresses learned from TRILL Data packet decapsulation in MAC-Reachability TLV as per [RFC6165] and advertise them using the ESADI protocol.

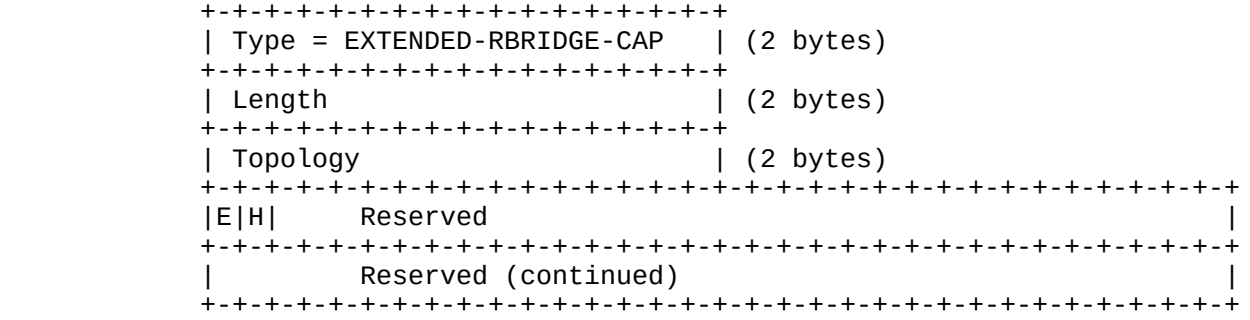
When the remote RBridge learns MAC addresses contained in the AA-LAALP-GROUP-MAC APPsub-TLV via the ESADI protocol [RFC7357], it sends the packets destined to these MAC addresses to the closest one (the one to which the remote RBridge has the least cost forwarding path) of those RBridges in the AAE identified by the LAALP ID in the AA-LAALP-GROUP-MAC APPsub-TLV. If there are multiple equal least cost member RBridges, the ingress RBridge is required to select a unique one in a pseudo-random way as specified in Section 5.3 of [RFC7357].

When another RBridge in the same AAE group receives an ESADI-LSP with the AA-LAALP-GROUP-MAC APPsub-TLV, it also learns MAC addresses of those end-nodes served by the corresponding LAALP. These MAC addresses SHOULD be learned as if those end-nodes are locally attached to this RBridge itself.

An AAE member RBridge MUST use the AA-LAALP-GROUP-MAC APPsub-TLV to advertise in ESADI the MAC addresses learned from a plain local link (a non LAALP link) with Data Labels that happen to be covered by the Data Labels of any attached LAALP. The reason is that MAC learning from TRILL Data packet decapsulation within these Data Labels at the remote edge RBridge has normally been disabled for this RBridge.

4.2. Extended RBridge Capability Flags APPsub-TLV

The following Extended RBridge Capability Flags APPsub-TLV will be included in an E-L1FS FS-LSP fragment zero [RFC7180bis] as an APPsub-TLV of the TRILL GENINFO-TLV.



- Type: Extended RBridge Capability (TRILL APPsub-TLV type tbd2)
- Length: Set to 8.
- Topology: Indicates the topology to which the capabilities apply. When this field is set to zero, this implies that the capabilities apply to all topologies or topologies are not in use [TRILL-MT].
- E: Bit 0 of the capability bits. When this bit is set, it indicates the originating IS acts as specified in Option B above.
- H: Bit 1 of the capability bits. When this bit is set, it indicates that the originating IS keeps multiple MAC attachments learned from TRILL Data packet decapsulation with fast path hardware, that is, it acts as specified in Option A above.
- Reserved: Flags extending from bit 2 through bit 63 of the capability fits reserved for future use. These MUST be sent as zero and ignored on receipt.

The Extended RBridge Capability Flags TRILL APPsub-TLV is used to notify other RBridges whether the originating IS supports the capability indicated by the E and H bits. For example, if E bit is set, it indicates the originating IS will act as defined in Option B. That is, it will disable the MAC learning from TRILL Data packet decapsulation within Data Labels advertised by AAE RBridges while waiting for the TRILL ESADI-LSPs to distribute the {MAC, Nickname, Data Label} association. Meanwhile, this RBridge is able to act as an AAE RBridge. It’s required to advertise MAC addresses learned from local LAALPs in TRILL ESADI-LSPs using the AA-LAALP-GROUP-MAC APPsub-

TLV defined in Section 4.1. If an RBridge in an AAE group, as specified herein, observe a remote RBridge interested in one or more of that AAE group's Data Labels, and the remote RBridge does not support, as indicated by its extended capabilities, either Option A or Option B, then the AAE group MUST fall back to active-standby mode.

5. Meeting the Design Goals

How this specification meets the major design goals of AAE is explored in this section.

5.1. No MAC Flip-Flopping (Normal Unicast Egress)

Since all RBridges talking with the AAE RBridges in the campus are able to see multiple locations for one MAC address in ESADI [RFC7357], a MAC address learned from one AAE member will not be overwritten by the same MAC address learned from another AAE member. Although multiple entries for this MAC address will be created, for return traffic the remote RBridge is required to adhere to a unique one of the locations (see Section 4.1) for each MAC address rather than keep flip-flopping among them.

5.2. Regular Unicast/Multicast Ingress

LAALP guarantees that each frame will be sent upward to the AAE via exactly one uplink. RBridges in the AAE simply follow the process per [RFC6325] to ingress the frame. For example, each RBridge uses its own nickname as the ingress nickname to encapsulate the frame. In such a scenario, each RBridge takes for granted that it is the Appointed Forwarder for the VLANs enabled on the uplink of the LAALP.

5.3. Correct Multicast Egress

A fundamental design goal of AAE is that there must be no duplication or forwarding loop.

5.3.1. No Duplication (Single Exit Point)

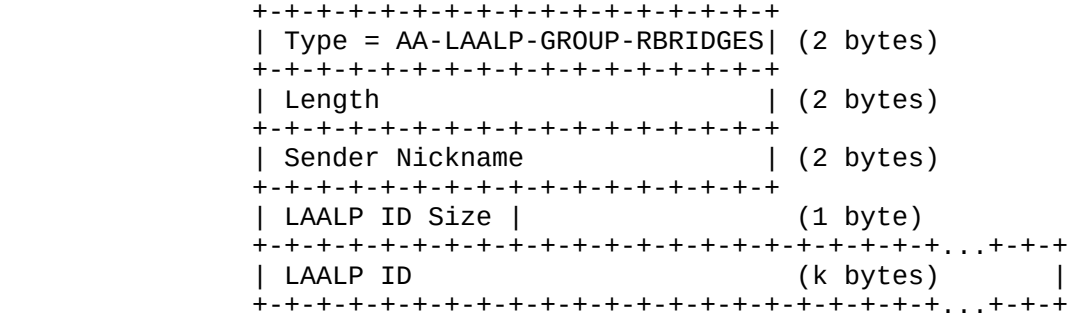
When multi-destination TRILL Data packets for a specific Data Label are received from the campus, it's important that exactly one RBridge out of the AAE group let through each multi-destination packet so no duplication will happen. The LAALP will have defined its selection function (using hashing or election algorithm) to designated a forwarder for a multi-destination frame. Since AAE member RBridges support the LAALP, they are able to utilize that selection function to determine the single exit point. If the output of the selection function points to the port attached to the receiving RBridge itself

(i.e., the packet should be egressed out of this node), it MUST egress this packet for that AAE group. Otherwise, the packet MUST NOT be egressed for that AAE group. (It is output or not as specified in [RFC6325] updated by [RFC7172] for ports that lead to non-AAE links.)

5.3.2. No Echo (Split Horizon)

When a multi-destination frame originated from an LAALP is ingressed by an RBridge of an AAE group, distributed to the TRILL network and then received by another RBridge in the same AAE group, it is important that this RBridge does not egress this frame back to this LAALP. Otherwise, it will cause a forwarding loop (echo). The well known ’split horizon’ technique is used to eliminate the echo issue.

RBridges in the AAE group need to split horizon based on the ingress RBridge nickname plus the VLAN of the TRILL Data packet. They need to set up per port filtering lists consisting of the tuple of <ingress nickname, VLAN>. Packets with information matching with any entry of the filtering list MUST NOT be egressed out of that port. The information of such filters is obtained by listening to the following "LAALP Group RBridges" APPsub-TLV included in the TRILL GENINFO TLV in FS-LSPs [RFC7180bis].



- Type: AA LAALP Grouped RBridges (TRILL APPsub-TLV type tbd3)
- Length: 3+k
- Sender Nickname: The nickname the originating IS will use as the ingress nickname. This field is useful because the originating IS might own multiple nicknames.
- LAALP ID Size: The length k of the LAALP ID in bytes.
- LAALP ID: The ID of the LAALP which is k bytes long. If the LAALP is an MC-LAG or DRNI, it is the 8-byte ID specified in Clause 6.3.2 in [802.1AX].

All enabled VLANs MUST be consistent on all ports connected to an LAALP. So the enabled VLANs need not be included in the AA-LAALP-GROUP-RBRIDGES TRILL APPsub-TLV. They can be locally obtained from the port attached to that LAALP.

Through parsing AA-LAALP-GROUP-RBRIDGES TRILL APPsub-TLVs, the receiving RBridge discovers all other RBridges connected to the same LAALP. The Sender Nickname of the originating IS will be added into the filtering list of the port attached to the LAALP. For example, RB3 in Figure 3.1 will set up a filtering list that looks like {<RB1, VLAN10>, <RB2, VLAN10>} on its port attached to LAALP1. According to split horizon, TRILL Data packets within VLAN10 ingressed by RB1 or RB2 will not be egressed out of this port.

When there are multiple LAALPs connected to the same RBridge, these LAALPs may have VLANs that overlap. Here a VLAN overlaps means this VLAN ID is enabled by multiple LAALPs. Customer may need hosts within these overlapped VLANs to communicate with each other. In Appendix A, several scenarios are given to explain how hosts communicate within the overlapped VLANs and how split horizon happens.

5.4. No Black-hole or Triangular Forwarding

If a sub-link of the LAALP fails while remote RBridges continue to send packets towards the failed port, a black-hole happens. If the AAE member RBridge with that failed port starts to redirect the packets to other member RBridges for delivery, triangular forwarding occurs.

The member RBridge attached to the failed sub-link makes use of the ESADI protocol to flush those failure affected MAC addresses as defined in Section 5.2 of [RFC7357]. After doing that, no packets will be sent towards the failed port, hence no black-hole will happen. Nor will the member RBridge need to redirect packets to other member RBridges, which may otherwise lead to triangular forwarding.

5.5. Load Balance Towards the AAE

Since a remote RBridge can see multiple attachments of one MAC address in ESADI, this remote RBridge can choose to spread the traffic towards the AAE members on a per flow basis. Each of them is able to act as the egress point. In doing this, the forwarding paths need not be limited to the least cost Equal Cost Multiple Paths from the ingress RBridge to the AAE RBridges. The traffic load from the remote RBridge towards the AAE RBridges can be balanced based on a pseudo-random selection method (see Section 4.1).

Note that the load balance method adopted at a remote ingress RBridge

is not to replace the load balance mechanism of LAALP. These two load spreading mechanisms should take effect separately.

5.6. Scalability

With option A, multiple attachments need to be recorded for a MAC address learned from AAE RBridges. More entries may be consumed in the MAC learning table. However, MAC addresses attached to an LAALP are usually only a small part of all MAC addresses in the whole TRILL campus. As a result, the extra space required by the multi-attached MAC addresses can usually be accommodated by RBridges unused MAC table space.

With option B, remote RBridges will keep the multiple attachments of a MAC address in the ESADI link state databases that are usually maintained by software. While in the MAC table that is normally implemented in hardware, an RBridge still establishes only one entry for each MAC address.

6. E-L1FS Backwards Compatibility

The Extended TLVs defined in Section 4 and 5 are to be used in an Extended Level 1 Flooding Scope (E-L1FS [RFC7356] [RFC7180bis]) PDU. For those RBridges that do not support E-L1FS, the EXTENDED-RBRIDGE-CAP TRILL APPsub-TLV will not be sent out either, and MAC multi-attach active-active is not supported.

7. Security Considerations

Authenticity for contents transported in IS-IS PDUs is enforced using regular IS-IS security mechanism [ISIS][RFC5310].

For security considerations pertaining to extensions transported by TRILL ESADI, see the Security Considerations section in [RFC7357].

For general TRILL security considerations, see [RFC6325].

8. IANA Considerations

8.1. TRILL APPsub-TLVs

IANA is requested to allocate three new types under the TRILL GENINFO TLV [RFC7357] for the TRILL APPsub-TLVs defined in Section 4.1, 4.2 and 5.3.2 of this document. The following entries are added to the "TRILL APPsub-TLV Types under IS-IS TLV 251 Application Identifier 1" Registry on the TRILL Parameters IANA web page.

Type	Name	Reference

tbd1(252)	AA-LAALP-GROUP-MAC	[This document]
tbd2(253)	EXTENDED-RBRIDGE-CAP	[This document]
tbd3(254)	AA-LAALP-GROUP-RBRIDGES	[This document]

8.2. Extended RBridge Capabilities Registry

IANA is requested to create a registry under the TRILL Parameters registry as follows:

Name: Extended RBridge Capabilities

Registration Procedure: Expert Review

Reference: [this document]

Bit	Mnemonic	Description	Reference
0	E	Option B Support	[this document]
1	H	Option A Support	[this document]
2-63	-	Unassigned	

8.3. Active Active Flags

IANA is requested to allocate two flag bits, with mnemonic "AA", as follows:

One flag bit appears in the "Interested VLANs and Spanning Tree Roots Sub-TLV".

Bit	Mnemonic	Description	Reference
0	M4	IPv4 Multicast Router Attached	[RFC7176]
1	M6	IPv6 Multicast Router Attached	[RFC7176]
2	-	Unassigned	
3	ES	ESADI Participation	[RFC7357]
4-15	-	(used for a VLAN ID)	[RFC7176]
16	AA	Enabled VLANs for Active-Active	[This document]
17-19	-	Unassigned	
20-31	-	(used for a VLAN ID)	[RFC7176]

One flag bit appears in the "Interested Labels and Spanning Tree Roots Sub-TLV".

Bit	Mnemonic	Description	Reference
0	M4	IPv4 Multicast Router Attached	[RFC7176]
1	M6	IPv6 Multicast Router Attached	[RFC7176]
2	BM	Bit Map	[RFC7176]
3	ES	ESADI Participation	[RFC7357]
4	AA	FGLs for Active-Active	[This document]
5-7	-	Unassigned	

9. Acknowledgements

Authors would like to thank the comments and suggestions from Andrew Qu, Donald Eastlake, Erik Nordmark, Fangwei Hu, Liang Xia, Weiguo Hao, Yizhou Li and Mukhtiar Shaikh.

10. References

10.1. Normative References

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

[RFC6165] Banerjee, A. and D. Ward, "Extensions to IS-IS for Layer-2 Systems", RFC 6165, April 2011.

[RFC6325] Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

[RFC6439] Perlman, R., Eastlake, D., Li, Y., Banerjee, A., and F. Hu, "Routing Bridges (RBridges): Appointed Forwarders", RFC 6439, November 2011.

[RFC7172] D. Eastlake 3rd and M. Zhang and P. Agarwal and R. Perlman and D. Dutt, "Transparent Interconnection of Lots of Links (TRILL): Fine-Grained Labeling", RFC 7172, May 2014.

[RFC7176] D. Eastlake 3rd and T. Senevirathne and A. Ghanwani and D. Dutt and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", RFC7176, May 2014.

[RFC7177] D. Eastlake 3rd and R. Perlman and A. Ghanwani and H. Yang and V. Manral, "Transparent Interconnection of Lots of Links (TRILL): Adjacency", RFC 7177, May 2014.

[RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, September 2014.

[RFC7357] Zhai, H., Hu, F., Perlman, R., Eastlake 3rd, D., and O. Stokes, "Transparent Interconnection of Lots of Links (TRILL): End Station Address Distribution Information (ESADI) Protocol", RFC 7357, September 2014.

[RFC7180bis] D. Eastlake, M. Zhang, et al, "TRILL: Clarifications, Corrections, and Updates", draft-ietf-trill-rfc7180bis, work in progress.

[802.1AX] IEEE, "IEEE Standard for Local and Metropolitan Area Networks - Link Aggregation", 802.1AX-2014, 24 December 2014.

10.2. Informative References

[RFC7379] Li, Y., Hao, W., Perlman, R., Hudson, J., and H. Zhai, "Problem Statement and Goals for Active-Active Connection at the Transparent Interconnection of Lots of Links (TRILL) Edge", RFC 7379, October 2014.

[PN] H. Zhai, T. Senevirathne, et al, "TRILL: Pseudo-Nickname for Active-active Access", draft-ietf-trill-pseudonode-nickname, work in progress.

[TRILL-MT] D. Eastlake, M. Zhang, A. Banerjee, V. Manral, "TRILL: Multi-Topology", draft-eastlake-trill-multi-topology, work in progress.

[ISIS] ISO, "Intermediate system to Intermediate system routeing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

Appendix A. Scenarios for Split Horizon

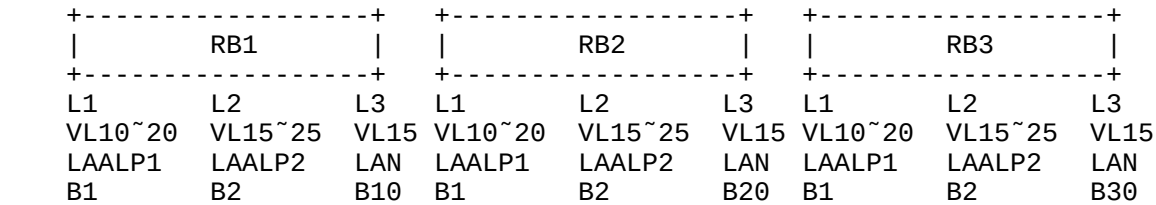


Figure A.1: An example topology to explain split horizon

Suppose RB1, RB2 and RB3 are the Active-Active group connecting LAALP1 and LAALP2. LAALP1 and LAALP2 are connected to B1 and B2 at their other ends. Suppose all these RBridges use port L1 to connect LAALP1 while they use port L2 to connect LAALP2. Assume all three L1 enable VLAN 10~20 while all three L2 enable VLAN 15~25. So that there is an overlap of VLAN 15~20. The customer needs hosts in these overlapped VLANs to communicate with each other. That is, hosts attached to B1 in VLAN 15~20 need to communicate with hosts attached to B2 in VLAN 15~20. Assume the remote plain RBridge RB4 also has hosts attached in VLAN 15~20 which need to communicate with those hosts in these VLANs attached to B1 and B2.

Two major requirements:

1. Frames ingressed from RB1-L1-VLAN 15~20 MUST NOT be egressed out of ports RB2-L1 and RB3-L1. At the same time,

2. frames coming from B1-VLAN 15~20 should reach B2-VLAN 15~20.

RB3 stores the information for split horizon on its ports L1 and L2. On L1: {<ingress_nickname_RB1, VLAN 10~20>, <ingress_nickname_RB2, VLAN 10~20>} and on L2: {<ingress_nickname_RB1, VLAN 15~25>, <ingress_nickname_RB2, VLAN 15~25>}.

Five clarification scenarios:

a. Suppose RB2/RB3 receives a TRILL multi-destination data packet with VLAN 15 and ingress nickname RB1. RB3 is the single exit point (selected out according to the hashing function of LAALP) for this packet. On ports L1 and L2, RB3 has covered <ingress_nickname_RB1, VLAN 15>, so that RB3 will not egress this packet out of either L1 or L2. Here, _split horizon_ happens.

   Beforehand, RB1 obtains a native frame on port L1 from B1 in VLAN 15. RB1 judges it should be forwarded as a multi-destination packet across the TRILL campus. Also, RB1 replicates this frame without TRILL encapsulation and sends it out of port L2, so that B2 will get this frame.

b. Suppose RB2/RB3 receives a TRILL multi-destination data packet with VLAN 15 and ingress nickname RB4. RB3 is the single exit point. On ports L1 and L2, since RB3 has not stored any tuple with ingress_ nickname_RB4, RB3 will decapsulate the packet and egress it out of both ports L1 and L2. So both B1 and B2 will receive the frame.

c. Suppose there is a plain LAN link port L3 on RB1, RB2 and RB3, connecting to B10, B20 and B30 respectively. These L3 ports happen

to be configured with VLAN 15. On port L3, RB2 and RB3 stores no information of split horizon for AAE (since this port has not been configured to be in any LAALP). They will egress the packet ingressed from RB1-L1 in VLAN 15.

d. If a packet is ingressed from RB1-L1 or RB1-L2 with VLAN 15, port RB1-L3 will not egress packets with ingress-nickname-RB1. RB1 needs to replicate this frame without encapsulation and sends it out of port L3. This kind of 'bounce' behavior for multi-destination frames is just as specified in paragraph 2 of Section 4.6.1.2 of [RFC6325].

e. If a packet is ingressed from RB1-L3, since RB1-L1 and RB1-L2 cannot egress packets with VLAN 15 and ingress-nickname-RB1, RB1 needs to replicate this frame without encapsulation and sends it out of port L1 and L2. (Also see paragraph 2 of Section 4.6.1.2 of [RFC6325].)

Author's Addresses

Mingui Zhang
Huawei Technologies
No.156 Beiqing Rd. Haidian District,
Beijing 100095 P.R. China

EMail: zhangmingui@huawei.com

Radia Perlman
EMC
2010 256th Avenue NE, #200
Bellevue, WA 98007 USA

EMail: radia@alum.mit.edu

Hongjun Zhai
Jinling Institute of Technology
99 Hongjing Avenue, Jiangning District
Nanjing, Jiangsu 211169 China

EMail: honjun.zhai@tom.com

Muhammad Durrani
Brocade
130 Holger Way
San Jose, CA 95134

EMail: mdurrani@brocade.com

Sujay Gupta
IP Infusion,
RMZ Centennial
Mahadevapura Post
Bangalore - 560048
India

EMail: sujay.gupta@ipinfusion.com

TRILL Working Group
Internet Draft
Intended status: Standard Track
Updates: 6325

Tissa Senevirathne
CISCO
Janardhanan Pathangi
DELL
Jon Hudson
Brocade

April 1, 2014

Expires: October 2014

Coordinated Multicast Trees (CMT) for TRILL
draft-ietf-trill-cmt-03.txt

Status of this Memo

Copyright Notice

Abstract

TRILL facilitates loop free connectivity to non-TRILL legacy networks via choice of an Appointed Forwarder for a set of VLANs. Appointed Forwarder provides load sharing based on VLAN with an active-standby model. Mission critical operations such as High Performance Data Centers require active-active load sharing model. The Active-Active load sharing model can be accomplished by representing any given non-TRILL legacy network with a single virtual RBridge. Virtual representation of the non-TRILL legacy network with a single RBridge poses serious challenges in multi-destination RPF (Reverse Path Forwarding) check calculations. This document specifies required enhancements to build Coordinated Multicast Trees (CMT) within the TRILL campus to solve related RPF issues. CMT provides flexibility to RBridges in selecting desired path of association to a given TRILL multi-destination distribution tree.

Table of Contents

1. Introduction

TRILL (Transparent Interconnection of Lots of Links) presented in [RFC6325] and other related documents, provides methods of utilizing all available paths for active forwarding, with minimum configuration. TRILL utilizes IS-IS (Intermediate System to Intermediate System [IS-IS]) as its control plane and uses a TRILL header with hop count.

[RFC6325], [6327bis] and [RFC6439] provide methods for interoperability between TRILL and Legacy networks. [RFC6439], provide an active-standby solution, where only one of the RBridges on a link with end stations is in the active forwarding state for end station traffic for any given VLAN. That RBridge is referred to as the Appointed Forwarder (AF). All frames ingressed into a TRILL network via the Appointed Forwarder are encapsulated with the TRILL header with a nickname held by the ingress AF RBridge. Due to failures, re-configurations and other network dynamics, the Appointed Forwarder for any set of VLANs may change. RBridges maintain forwarding tables that contain destination MAC address and VLAN to egress RBridge binding. In the event of AF change, forwarding tables of remote RBridges may continue to forward traffic to the previous AF and that traffic may get discarded at the egress, causing traffic disruption.

Mission critical applications such as High Performance Data Centers require resiliency during failover. The active-active forwarding model minimizes impact during failures and maximizes the available network bandwidth. A typical deployment scenario, depicted in Figure 1, which may have either End Stations and/or Legacy bridges attached to the RBridges. These Legacy devices typically are multi-homed to several RBridges and treat all of the uplinks as a single Multi-Chassis Link Aggregation (MC-LAG) bundle. The Appointed Forwarder designation presented in [RFC6439] requires each of the edge RBridges to exchange TRILL hello packets. By design, an MC-LAG does not forward packets received on one of the member ports of the MC-LAG to other member ports of the same MC-LAG. As a result the AF

designation methods presented in [RFC6439] cannot be applied to deployment scenario depicted in Figure 1.

An active-active load-sharing model can be implemented by representing the edge of the network connected to a specific edge group of RBridges by a single virtual RBridge. Each virtual RBridge MUST have a nickname unique within its TRILL campus. In addition to an active-active forwarding model, there may be other applications that may requires similar representations.

Sections 4.5.1 and 4.5.2 of [RFC6325] as updated by [clearcor] specify distribution tree calculation and RPF (Reverse Path Forwarding) check calculation algorithms for multi-destination forwarding. These algorithms strictly depend on link cost and parent RBridge priority. As a result, based on the network topology, it may be possible that a given edge RBridge, if it is forwarding on behalf of the virtual RBridge, may not have a candidate multicast tree that the edge RBridge can forward traffic on because there is no tree for which the virtual RBridge is a leaf node from the edge RBridge.

In this document we present a method that allows RBridges to specify the path of association for real or virtual child nodes to distribution trees. Remote RBridges calculate their forwarding tables and derive the RPF for distribution trees based on the distribution tree association advertisements. In the absence of distribution tree association advertisements, remote RBridges derive the SPF (Shortest Path First) based on the algorithm specified in section 4.5.1 of [RFC 6325].

Other applications, beside the above mentioned active-active forwarding model, may utilize the distribution tree association framework presented in this document to associate to distribution trees through a preferred path.

This proposal requires presence of multiple multi-destination trees within the TRILL campus and updating all the RBridges in the network to support the new Affinity sub-TLV (Section 3.). It is expected that both of these requirements will be met as they are control plane changes, and will be common deployment scenarios. In case either of the above two conditions are not met RBridges MUST support a fallback option for interoperability. Since the fallback is expected to be a temporary phenomenon till all RBridges are upgraded, this proposal gives guidelines for such fallbacks, and does not mandate or specify any specific set of fallback options.

1.1. Scope and Applicability

This document specifies an Affinity sub-TLV to solve associated RPF issues at the active-active edge. Specific methods in this document for making use of the Affinity sub-TLV are applicable where multiple RBridges are connected to an edge device through multi-chassis link aggregation or to a multiport server or some similar arrangement where the RBridges cannot see each other's Hellos.

This document DOES NOT provide other required operational elements to implement active-active edge solution, such as methods of multi-chassis link aggregation. Solution specific operational elements are outside the scope of this document and will be covered in solution specific documents. (See, for example [TRILLPN].)

Examples provided in this document are for illustration purposes only.

1.2. Contributors

The work in this document is a result of much passionate discussions and contributions from following individuals. Their names are listed in alphabetical order:

Ayan Banerjee, Dinesh Dutt, Donald Eastlake, Mingui Zhang, Radia Perlman, Sam Aldrin, Shivakumar Sundaram and Zhai Hongjun.

2. Conventions used in this document

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 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC2119] significance.

2.1. Acronyms

MC-LAG: . Multi-Chassis Link Aggregation is a solution specific extension to [8021AX], that facilitates connecting group of links from an originating device (A) to a group of discrete devices (B). Device (A) treats, all of the links in a given Multi-Chassis Link Aggregation bundle as a single logical interface and treats all devices in Group (B) as a single logical device for all forwarding purposes. Device (A) does not forward packets receive on Multi-

Chassis Link bundle out of the same Multi-Chassis link bundle. Figure 1 depicts a specific use case example.

CE : Classical Ethernet device, that is a device that performs forwarding based on 802.1Q bridging. This also can be end-station or a server.

RPF: Reverse Path Forwarding. See section 4.5.2 of [RFC6325].

3. The AFFINITY sub-TLV

Association of an RBridge to a multi-destination distribution tree through a specific path is accomplished by using a new IS-IS sub-TLV, the Affinity sub-TLV.

The AFFINITY sub-TLV appears in Router capability TLVs that are within LSP PDUs, as described in [6326bis] which specifies the code point and data structure for the Affinity sub-TLV.

4. Multicast Tree Construction and Use of Affinity Sub-TLV

Figure 1 and Figure 2 below show the reference topology and a logical topology using CMT to provide active-active service.

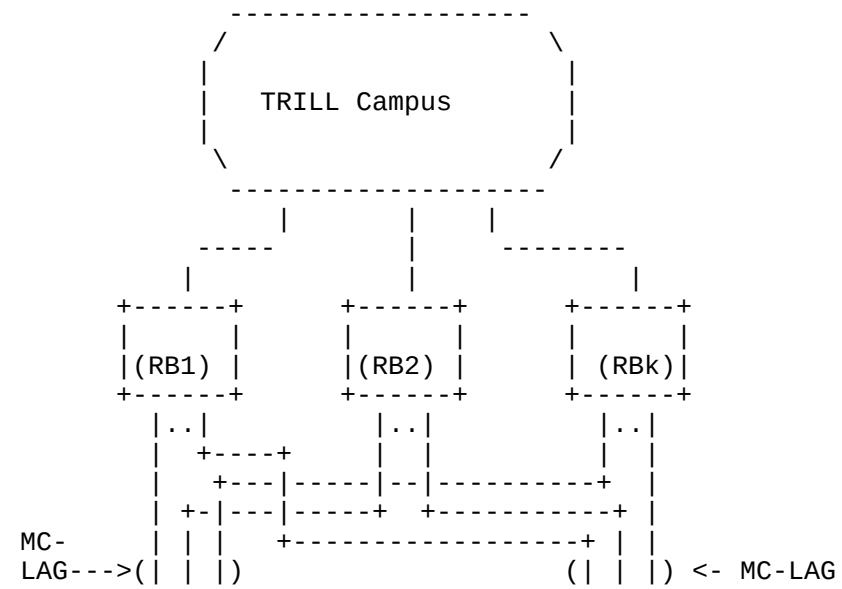


Figure 1 Reference Topology

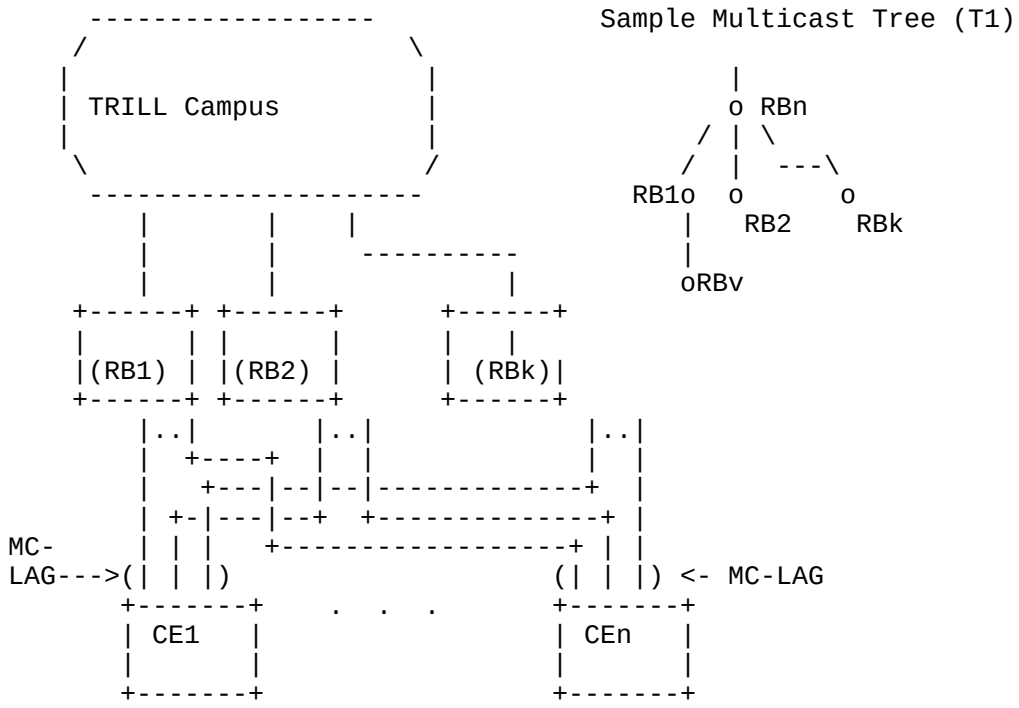


Figure 2 Example Logical Topology

4.1. Update to RFC 6325

Section 4.5.1 of [RFC6325], is updated as below:

Each RBridge that desires to be the parent RBridge for child Rbridge RBy in a multi-destination distribution tree x announces the desired association using an Affinity sub-TLV. The child RBridge RBy is specified by its nickname (or one of its nicknames if it holds more than one).

When such an Affinity sub-TLV is present, the association specified by the affinity sub-TLV MUST be used when constructing the multi destination distribution tree except in case of conflicting Affinity sub-TLV which are resolved as specified in Section 5.3. In the absence of such an Affinity sub-TLV, or if there are any RBridges in the campus that are do not support Affinity sub-TLV, distribution trees tree are calculated as specified in the section 4.5.1 of [RFC6325] as updated by [clearcor]. Section 4.3. below specifies how to identify RBridges that support Affinity sub-TLV capability.

4.2. Announcing virtual RBridge nickname

Each edge RBridge RB1 to RBk advertises in its LSP virtual RBridge nickname RBv using the Nickname sub-TLV (6), [6326bis], along with their regular nickname or nicknames.

It will be possible for any RBridge to determine that RBv is a virtual RBridge because each RBridge (RB1 to RBk) this appears to be advertising that it is holding RBv is also advertising an Affinity sub-TLV asking that RBv be its child in one or more trees.

Virtual RBridges are ignored when determining the distribution tree roots for the campus.

All RBridges outside the edge group assume that multi-destination packets with ingress nickname RBv might use any of the distribution trees that any member of the edge group is advertising that it might use.

4.3. Affinity Sub-TLV Capability.

RBridges that announce the TRILL version sub-TLV [6326bis] and set the Affinity capability bit (Section 7.) support the Affinity sub-TLV and calculation of multi-destination distribution trees and RPF checks as specified herein.

5. Theory of operation

5.1. Distribution Tree provisioning

Let’s assume there are n distribution trees and k edge RBridges in the edge group of interest.

If n >= k

Let’s assume edge RBridges are sorted in numerically ascending order by SystemID such that RB1 < RB2 < RBk. Each Rbridge in the numerically sorted list is assigned a monotonically increasing number j such that; RB1=0, RB2=1, RBi=j and RBi+1=j+1.

Assign each tree to RBi such that tree number { (tree_number) % k}+1 is assigned to RBridge i for tree_number from 1 to n. where n is the number of trees and k is the number of RBridges considered for tree allocation.

If n < k

Distribution trees are assigned to RBridges RB1 to RBn, using the same algorithm as n >= k case. RBridges RBn+1 to RBk do not participate in active-active forwarding process on behalf of RBv.

5.2. Affinity Sub-TLV advertisement

Each RBridge in the RB1..RBk domain advertises an Affinity TLV for RBv to be its child.

As an example, let’s assume that RB1 has chosen Trees t1 and tk+1 on behalf of RBv.

RB1 advertises affinity TLV; {RBv, Num of Trees=2, t1, tk+1.

Other RBridges in the RB1..RBk edge group follow the same procedure.

5.3. Affinity sub-TLV conflict resolution

In TRILL, multi-destination distribution trees are built outward from the root. If an RBridges RB1 advertises an Affinity sub-TLV with an AFFINITY RECORD that asks for RBridge RBroot to be its child in a tree rooted at RBroot, that AFFINITY RECORD is in conflict with TRILL distribution tree root determination and MUST be ignored.

If an RBridge RB1 advertises an Affinity sub-TLV with an AFFINITY RECORD that's ask for nickname RBn to be its child in any tree and RB1 is not adjacent to a real or virtual RBridge RBn, that AFFINITY RECORD is in conflict with the campus topology and MUST be ignored.

If different RBridges advertise Affinity sub-TLVs that try to associate the same virtual RBridge as their child in the same tree or trees, those Affinity sub-TLVs are in conflict for those trees. The nicknames of the conflicting RBridges are compared to identify which RBridge holds the nickname that is the highest priority to be a tree root, with the System ID as the tie breaker

The RBridge with the highest priority to be a tree root will retain the Affinity association. Other RBridges with lower priority to be a tree root MUST stop advertising their conflicting Affinity sub-TLV, re-calculate the multicast tree affinity allocation, and, if appropriate, advertise a new non-conflict Affinity sub-TLV.

Similarly, remote RBridges MUST honor the Affinity sub-TLV from the RBridge with the highest priority to be a tree root (use system-ID as the tie-breaker in the event of conflicting priorities) and ignore the conflicting Affinity sub-TLV entries advertised by the RBridges with lower priorities to be tree roots.

5.4. Ingress Multi-Destination Forwarding

If there is at least one tree on which RBv has affinity via RBk, then RBk performs the following operations, for multi-destination frames received from a CE node:

1. Flood to locally attached CE nodes subjected to VLAN and multicast pruning.
2. Ingress in the TRILL header and assign ingress RBridge nickname as RBv. (nickname of the virtual RBridge).
3. Forward to one of the distribution trees, tree x in which RBv is associated with RBk

5.4.1. Forwarding when n < k

If there is no tree on which RBv can claim affinity via RBk (Probably because the number of trees n built is less than number of RBridges k announcing the affinity sub-TLV), then RBk MUST fall back to one of the following

1. This RBridge should stop forwarding frames from the CE nodes, and should mark that port as disabled. This will prevent CE

nodes from forwarding data on to this RBridge, and only use those RBridges which have been assigned a tree -
-OR-

2. This RBridge tunnels multi-destination frames received from attached native devices to an RBridge RBy that has an assigned tree. The tunnel destination should forward it to the TRILL network, and also to its local access links. (The mechanism of tunneling and handshake between the tunnel source and destination are out of scope of this specification and may be addressed in future documents).

Above fallback options may be specific to active-active forwarding scenario. However, as stated above, Affinity sub-TLV may be used in other applications. In such event the application SHOULD specify applicable fallback options.

5.5. Egress Multi-Destination Forwarding

5.5.1. Traffic Arriving on an assigned Tree to RBk-RBv

Multi-destination frames arriving at RBk on a Tree x, where RBk has announced the affinity of RBv via x, MUST be forwarded to CE members of RBv that are in the frame’s VLAN. Forwarding to other end-nodes and RBridges that are not part of the network represented by the RBv virtual RBridge MUST follow the forwarding rules specified in [RFC6325].

5.5.2. Traffic Arriving on other Trees

Multi-destination frames arriving at RBk on a Tree y, where RBk has not announced the affinity of RBv via y, MUST NOT be forwarded to CE members of RBv. Forwarding to other end-nodes and RBridges that are not part of the network represented by the RBv virtual RBridge MUST follow the forwarding rules specified in RFC6325.

5.6. Failure scenarios

The below failure recovery algorithm is presented only as a guideline. Implementations MAY include other failure recover algorithms. Details of such algorithms are outside the scope of this document.

5.6.1. Edge RBridge RBk failure

Each of the member RBridges of given virtual RBridge edge group is aware of its member RBridges through configuration or some other method.

Member RBridges detect nodal failure of a member RBridge through IS-IS LSP advertisements or lack thereof.

Upon detecting a member failure, each of the member RBridges of the RBv edge group start recovery timer T_rec for failed RBridge RBi. If the previously failed RBridge RBi has not recovered after the expiry of timer T_rec, members RBridges perform distribution tree assignment algorithm specified in section 5.1. Each of the member RBridges re-advertises the Affinity sub-TLV with new tree assignment. This action causes the campus to update the tree calculation with the new assignment.

RBi upon start-up, starts advertising its presence through IS-IS LSPs and starts a timer T_i. Member RBridges detecting the presence of RBi start a timer T_j. Timer T_j SHOULD be at least < T_i/2. (Please see note below)

Upon expiry of timer T_j, member RBridges recalculate the multi-destination tree assignment and advertised the related trees using Affinity sub-TLV.

Upon expiry of timer T_i, RBi recalculate the multi-destination tree assignment and advertises the related trees using Affinity TLV.

Note: Timers T_i and T_j are designed so as to minimize traffic down time and avoid multi-destination packet duplication.

5.7. Backward compatibility

Implementations MUST support backward compatibility mode to interoperate with pre Affinity sub-TLV RBRidges in the network. Such backward compatibility operation MAY include, however is not limited to, tunneling and/or active-standby modes of operations.

Example:

Step 1. Stop using virtual RBridge nickname for traffic ingressing from CE nodes
Step 2. Stop performing active-active forwarding. And fall back to active standby forwarding, based on locally defined policies. Definition of such policies is outside the scope of this document and may be addressed in future documents.

6. Security Considerations

In general, the RBridges in a campus are trusted routers and the authenticity of their link state information (LSPs) and link local

PDUs (Hellos, etc.) can be enforced using regular IS-IS security mechanisms [IS-IS] [RFC5410]. This including authenticating the contents of the PDUs used to transport Affinity sub-TLVs.

The particular Security Considerations involve with different applications of the Affinity sub-TLV will be covered in the document(s) specifying those applications.

For general TRILL Security Considerations, see [RFC6325].

7. IANA Considerations

IANA is requested to allocate a capability bit for ''Affinity Supported'' in the TRILL-VER sub-TLV. "Affinity Supported" capability bit and Affinity sub-TLV are specified and allocated in [6326bis].

8. References

8.1. Normative References

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

[RFC5310] Bhatia, M., et.al. ''IS-IS Generic Cryptographic Authentication'', RFC 5310, February 2009.

[RFC6325] Perlman, R., et.al. ''RBridge: Base Protocol Specification'', RFC 6325, July 2011.

[6327bis] Eastlake, D. et.al., ''RBridge: Adjacency'', draft-eastlake-trill-rfc6327bis, Work in Progress, July 2011.

[RFC6439] Eastlake, D. et.al., ''RBridge: Appointed Forwarder'', RFC 6439, November 2011.

[6326bis] Eastlake, D. et.al., ''Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS'', draft-eastlake-isis-rfc6326bis, Work in Progress, December 2011.

[clearcor] Eastlake, D. et.al., ''TRILL: Clarifications, Corrections, and Updates'', draft-ietf-trill-clear-correct, Work in Progress, July 2011.

[IS-IS] ISO/IEC, ''Intermediate System to Intermediate System Routing Information Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)'' ISO/IEC 10589:2002.

8.2. Informative References

[RFC6165] Banerjee, A. and Ward, D. ''Extensions to IS-IS for Layer-2 Systems'', RFC 6165, April 2011.

[RFC4971] Vasseur, JP. et.al ''Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information'', RFC 4971, July 2007.

[TRILLPN] Zhai,H., et.al ''RBridge: Pseudonode Nickname'', draft-hu-trill-pseudonode-nickname, Work in progress, November 2011.

[8021AX] IEEE, ''Link Aggregration'', IEEE Std 802.1AX-2008, November 2008.

[8021Q] IEEE, ''Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks'', IEEE Std 802.1Q-2011, August, 2011

9. Acknowledgments

Authors wish to extend their appreciations towards individuals who volunteered to review and comment on the work presented in this document and provided constructive and critical feedback. Specific acknowledgements are due for Anoop Ghanwani, Ronak Desai, and Varun Shah. Very special Thanks to Donald Eastlake for his careful review and constructive comments.

This document was prepared using 2-Word-v2.0.template.dot.

Appendix A. Change History.

From -01 to -02:

Replaced all references to ''LAG'' with references to Multi-Chassis (MC-LAG) or the like.

Expanded, Security Considerations section.

Other editorial changes.

From -02 to -03

Minor editorial changes

Authors' Addresses

Tissa Senevirathne
Cisco Systems
375 East Tasman Drive,
San Jose, CA 95134

Phone: +1-408-853-2291
Email: tsenevir@cisco.com

Janardhanan Pathangi
Dell/Force10 Networks
Olympia Technology Park,
Guindy Chennai 600 032

Phone: +91 44 4220 8400
Email: Pathangi_Janardhanan@Dell.com

Jon Hudson
Brocade
130 Holger Way
San Jose, CA 95134 USA

Email: jon.hudson@gmail.com

INTERNET-DRAFT
Intended status: Proposed Standard
Updates: ESADI

Linda Dunbar
Donald Eastlake
Huawei
Radia Perlman
Intel
Igor Gashinsky
Yahoo
Yizhou Li
Huawei

Expires: August 13, 2014
February 14, 2014

TRILL: Edge Directory Assist Mechanisms
<draft-ietf-trill-directory-assist-mechanisms-00.txt>

Abstract

This document describes mechanisms for providing directory service to TRILL (Transparent Interconnection of Lots of Links) edge switches. The directory information provided can be used in reducing multi-destination traffic, particularly ARP/ND and unknown unicast flooding.

Status of This Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Distribution of this document is unlimited. Comments should be sent to the TRILL working group mailing list.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html. The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Table of Contents

1. Introduction

[RFC7067] gives a problem statement and high level design for using directory servers to assist TRILL [RFC6325] edge nodes to reduce multi-destination ARP/ND and unknown unicast flooding traffic and to potentially improve security against address spoofing within a TRILL campus. Because multi-destination traffic becomes an increasing burden as a network scales up in number of nodes, reducing ARP/ND and unknown unicast flooding improves TRILL network scalability. This document describes specific mechanisms for directory servers to assist TRILL edge nodes. These mechanisms are optional to implement.

The information held by the Directory(s) is address mapping and reachability information. Most commonly, what MAC address [RFC7042] corresponds to an IP address within a Data Label (VLAN or FGL (Fine Grained Label [RFCfgl])) and the egress TRILL switch (RBridge) (and optionally what specific TRILL switch port) from which that MAC address is reachable. But it could be what IP address corresponds to a MAC address or possibly other address mappings or reachability.

In the data center environment, it is common for orchestration software to know and control where all the IP addresses, MAC addresses, and VLANs/tenants are in a data center. Thus such orchestration software is appropriate for providing the directory function or for supplying the Directory(s) with directory information.

Directory services can be offered in a Push or Pull Mode. Push Mode, in which a directory server pushes information to TRILL switches indicating interest, is specified in Section 2. Pull Mode, in which a TRILL switch queries a server for the information it wants, is specified in Section 3. More detail on modes of operation, including hybrid Push/Pull, are provided in Section 4.

The mechanisms used to initially populate directory data in primary servers is beyond the scope of this document. A primary server can use the Push Directory service to provide directory data to secondary servers as described in Section 2.5.

1.1 Terminology

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].

The terminology and acronyms of [RFC6325] are used herein along with the following:

COP: Complete Push flag bit. See Sections 2 and 8.1 below.

CSNP Time: Complete Sequence Number PDU Time. See ESDADI [RFCesadi] and Section 8.1 below.

Data Label: VLAN or FGL.

FGL: Fine Grained Label [RFCfgl].

Host: Application running on a physical server or a virtual machine. A host must have a MAC address and usually has at least one IP address.

IP: Internet Protocol. In this document, IP includes both IPv4 and IPv6.

PSH: Push Directory flag bit. See Sections 2 and 8.1 below.

PUL: Pull Directory flag bit. See Sections 3 and 8.3 below.

primary server: A Directory server that obtains the information it is serving up by a reliable mechanism outside the scope of this document designed to assure the freshness of that information. (See secondary server.)

RBridge: An alternative name for a TRILL switch.

secondary server: A Directory server that obtains the information it is serving up from one or more primary servers.

tenant: Sometimes used as a synonym for FGL.

TRILL switch: A device that implements the TRILL protocol.

2. Push Model Directory Assistance Mechanisms

In the Push Model [RFC7067], one or more Push Directory servers reside at TRILL switches and push down the address mapping information for the various addresses associated with end station interface and the TRILL switches from which those interfaces are reachable [IA]. This service is scoped by Data Label (VLAN or FGL [RFCfgl]). A Push Directory also advertises whether or not it believes it has pushed complete mapping information for a Data Label. It might be pushing only a subset of the mapping and/or reachability information for a Data Label. The Push Model uses the ESADI [RFCesadi] protocol as its distribution mechanism.

With the Push Model, if complete address mapping information for a Data Label being pushed is available, a TRILL switch (RBridge) which has that complete pushed information and is ingressing a native frame can simply drop the frame if the destination unicast MAC address can’t be found in the mapping information available, instead of flooding the frame (ingressing it as an unknown MAC destination TRILL Data frame). But this will result in lost traffic if ingress TRILL switch’s directory information is incomplete.

2.1 Requesting Push Service

In the Push Model, it is necessary to have a way for a TRILL switch to request information from the directory server(s). TRILL switches simply use the ESADI [RFCesadi] protocol mechanism to announce, in their core IS-IS LSPs, the Data Labels for which they are participating in ESADI by using the Interested VLANs and/or Interested Labels sub-TLVs [RFC6326bis]. This will cause them to be pushed the Directory information for all such Data Labels that are being served by one or more Push Directory servers.

2.2 Push Directory Servers

Push Directory servers advertise their availability to push the mapping information for a particular Data Label to each other and to ESADI participants for that Data Label through ESADI by turning on the a flag bit in their ESADI Parameter APPsub-TLV for that ESADI instance (see [RFCesadi] and Section 8.1). Each Push Directory server MUST participate in ESADI for the Data Labels for which it will push mappings and set the PSH (Push Directory) bit in its ESADI-Parameters APPsub-TLV for that Data Label.

For robustness, it is useful to have more than one copy of the data being pushed. Each Push Directory server is configured with a number

in the range 1 to 8, which defaults to 2, for each Data Label for which it can push directory information. If the Push Directories for a Data Label are configured the same in this regard and enough such servers are available, this is the number of copies of the directory that will be pushed.

Each Push Directory server also has an 8-bit priority to be Active (see Section 8.1 of this document). This priority is treated as an unsigned integer where larger magnitude means higher priority and is in its ESADI Parameter APPsub-TLV. In cases of equal priority, the 6-byte IS-IS System IDs of the tied Push Directories are used as a tie breaker and treated as an unsigned integer where larger magnitude means higher priority.

For each Data Label it can serve, each Push Directory server orders, by priority, the Push Directory servers that it can see in the ESADI link state database for that Data Label that are data reachable [RFCclear] and determines its own position in that order. If a Push Directory server is configured to believe that N copies of the mappings for a Data Label should be pushed and finds that it is number K in the priority ordering (where number 1 is highest priority and number K is lowest), then if K is less than or equal to N the Push Directory server is Active. If K is greater than N it is Passive. Active and Passive behavior are specified below.

For a Push Directory to reside on an end station, one or more TRILL switches locally connected to that end station must proxy for the Push Directory server and advertise themselves as Push Directory servers. It appears to the rest of the TRILL campus that these TRILL switches (that are proxying for the end station) are the Push Directory server(s). The protocol between such a Push Directory end station and the one or more proxying TRILL switches acting as Push Directory servers is beyond the scope of this document.

2.3 Push Directory Server State Machine

The subsections below describe the states, events, and corresponding actions for Push Directory servers.

2.3.1 Push Directory States

A Push Directory Server is in one of six states, as listed below, for each Data Label it can serve. In addition, it has an internal State-Transition-Time variable for each Data Label it can serve which is set at each state transition and which enables it to determine how long it has been in its current state for that Data Label.

Down: A completely shut down virtual state defined for convenience in specifying state diagrams. A Push Directory Server in this state does not advertise any Push Directory data. It may be participating in ESDADI [RFCesadi] with the PSH bit zero in its ESADI-Parameters or might be not participating in ESADI at all. All states other than the Down state are considered to be Up states.

Passive: No Push Directory data is advertised. Any outstanding EASDI-LSP fragments containing directory data are updated to remove that data and if the result is an empty fragment (contains nothing except possibly an Authentication TLV), the fragment is purged. The Push Directory participates in ESDADI [RFCesadi] and advertises its ESADI fragment zero that includes an ESADI-Parameters APPsub-TLV with the PSH bit set to one and COP (Complete Push) bit zero.

Active: If a Push Directory server is Active, it advertises its directory data and any changes through ESADI [RFCesadi] in its ESADI-LSPs using the Interface Addresses [IA] APPsub-TLV and updates that information as it changes. The PSH bit is set to one in the ESADI-Parameters and the COP bit set to zero.

Completing: Same behavior as the Active state but responds differently to events.

Complete: The same behavior as Active except that the COP bit in the ESADI-Parameters APPsub-TLV is set to one and the server responds differently to events.

Reducing: The same behavior as Complete but responds differently to events. The PSH bit remains a one but the COP bit is cleared to zero in the ESADI-Parameters APPsub-TLV. Directory updates continue to be advertised.

2.3.2 Push Directory Events and Conditions

Three auxiliary conditions referenced later in this section are defined as follows for convenience:

The Activate Condition: The Push Directory server determines that it is priority K among the data reachable Push Directory servers (where highest priority is 1), the server is configured that there should be N copies pushed, and K is less than or equal to N. For example, the Push Directory server is configured that 2 copies should be pushed and finds that it is priority 1 or 2 among the Push Directory servers it can see.

The Pacify Condition: The Push Directory server determines that it is priority K among the data reachable data reachable Push Directory servers (where highest priority is 1), the server is configured that there should be N copies pushed, and K is greater than N. For example, the Push Directory server is configured that 2 copies should be pushed and finds that it is priority 3 or lower priority (higher number) among the Push directory servers it can see.

The Time Condition: The Push Directory server has been in its current state for an amount of time equal to or larger than its CSNP time (see Section 8.1).)

The events and conditions listed below cause state transitions in Push Directory servers.

1. Push Directory server was Down but is now up.

2. The Push Directory server or the TRILL switch on which it resides is being shut down.

3. The Activate Condition is met and the server is not configured to believe it has complete data.

4. The server determines that the Pacify Condition is met.

5. The Activate Condition is met and the server is configured to believe it has complete data.

6. The server is configured to believe it does not have complete data.

7. The Time Condition is met.

2.3.3 State Transition Diagram and Table

The state transition table is as follows:

Event	Down	Passive	Active	Completing	Complete	Reducing
1	Passive	Passive	Active	Completing	Complete	Reducing
2	Down	Down	Passive	Passive	Reducing	Reducing
3	Down	Active	Active	Active	Reducing	Reducing
4	Down	Passive	Passive	Passive	Reducing	Reducing
5	Down	Completing	Complete	Completing	Complete	Complete
6	Down	Passive	Active	Active	Reducing	Reducing
7	Down	Passive	Active	Complete	Complete	Active

The above state table is equivalent to the following transition

diagram:

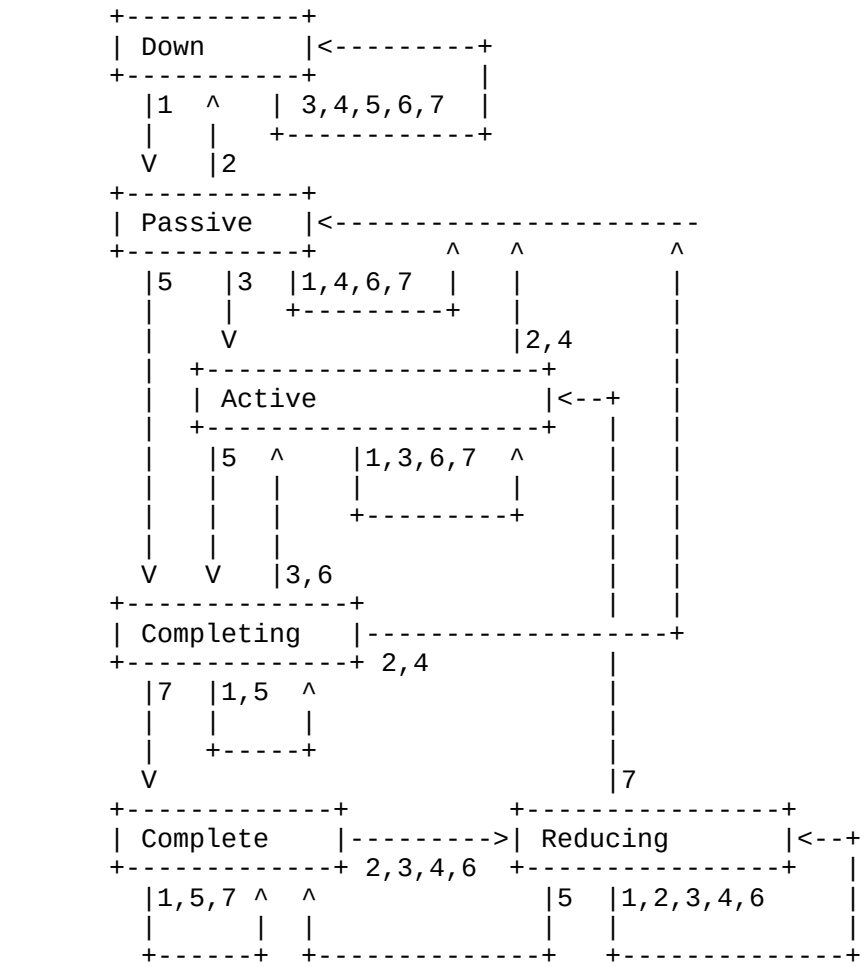


Figure 1. Push Server State Diagram

2.4 Additional Push Details

Push Directory mappings can be distinguished for other data distributed through ESADI because mappings are distributed only with the Interface Addresses APPsub-TLV [IA] and are flagged as being Push Directory data.

TRILL switches, whether or not they are a Push Directory server, MAY continue to advertise any locally learned MAC attachment information in ESDADI [RFCesadi] using the Reachable MAC Addresses TLV [RFC6165].

However, if a Data Label is being served by complete Push Directory servers, advertising such locally learned MAC attachment generally SHOULD NOT be done as it would not add anything and would just waste bandwidth and ESADI link state space. An exception might be when a TRILL switch learns local MAC connectivity and that information appears to be missing from the directory mapping.

Because a Push Directory server may need to advertise interest in Data Labels even if it does not want to receive end station multidestination data in those Data Labels, the No Data (NOD) flag bit is provided as specified in Section 8.3.

When a Push Directory server is no longer data reachable [RFCclear], TRILL switches MUST ignore any Push Directory data from that server because it is no longer being updated and may be stale.

The nature of dynamic distributed asynchronous systems is such that it is impossible for a TRILL switch receiving Push Directory information to be absolutely certain that it has complete information. However, it can obtain a reasonable assurance of complete information by requiring two conditions to be met:

1. The PSH and COP bits are on in the ESADI zero fragment from the server for the relevant Data Label.
2. It has had continuous data connectivity to the server for the larger of the client's and the server's CSNP times.

Condition 2 is necessary because a client TRILL switch might be just coming up and receive an EASDI LSP meeting the requirement in condition 1 above but have not yet received all of the ESADI LSP fragment from the Push Directory server.

There may be conflicts between mapping information from different Push Directory servers or conflicts between locally learned information and information received from a Push Directory server. In case of such conflicts, information with a higher confidence value [RFC6325] is preferred over information with a lower confidence. In case of equal confidence, Push Directory information is preferred to locally learned information and if information from Push Directory servers conflicts, the information from the higher priority Push Directory server is preferred.

2.5 Primary to Secondary Server Push Service

A secondary Push or Pull Directory server is one that obtains its data from a primary directory server. Other techniques MAY be used but, by default, this data transfer occurs through the primary server acting as a Push Directory server for the Data Labels involved while the secondary directory server takes the pushed data it receives from the highest priority Push Directory server and re-originates it. Such

a secondary server may be a Push Directory server or a Pull Directory server or both for any particular Data Label.

3. Pull Model Directory Assistance Mechanisms

In the Pull Model [RFC7067], a TRILL switch (RBridge) pulls directory information from an appropriate Directory Server when needed.

Pull Directory servers for a particular Data Label X are found by looking in the core TRILL IS-IS link state database for data reachable TRILL switches that advertise themselves by having the Pull Directory flag (PUL) on in their Interested VLANs or Interested Labels sub-TLV [RFC6326bis] for that Data Label. If multiple such TRILL switches indicate that they are Pull Directory Servers for a particular Data Label, pull requests can be sent to any one or more of them but it is RECOMMENDED that pull requests be preferentially sent to the server or servers that are lower cost from the requesting TRILL switch.

Pull Directory requests are sent by enclosing them in an RBridge Channel [Channel] message using the Pull Directory channel protocol number (see Section 8.2). Responses are returned in an RBridge Channel message using the same channel protocol number. See Section 3.2 for Query and Response message formats. For cache consistency or notification purposes, Pull Directory servers can sent unsolicited Update messages to client TRILL switches that believe may be holding old data and those clients can acknowledge such updates, as described in Section 3.3. All these messages have a common header as described in Section 3.1. Errors returns can be sent for queries or updates as described in Section 3.5.

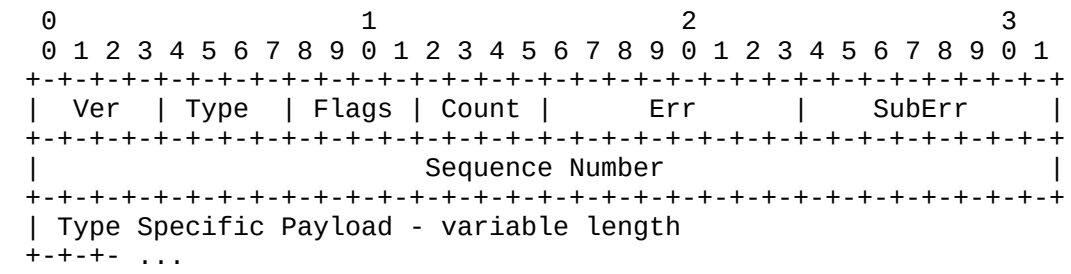
The requests to Pull Directory Servers are typically derived from ingressed ARP [RFC826], ND [RFC4861], or RARP [RFC903] messages, or data frames with unknown unicast destination MAC addresses, intercepted by an ingress TRILL switch as described in Section 4.

Pull Directory responses include an amount of time for which the response should be considered valid. This includes negative responses that indicate no data is available. Thus both positive responses with data and negative responses can be cached and used to locally handle ARP, ND, RARP, or unknown destination MAC frames, until the responses expire. If information previously pulled is about to expire, a TRILL switch MAY try to refresh it by issuing a new pull request but, to avoid unnecessary requests, SHOULD NOT do so if it has not been recently used. The validity timer of cached Pull Directory responses is NOT reset or extended merely because that cache entry is used.

3.1 Pull Directory Message Common Format

All Pull Directory messages are transmitted as the payload of RBridge Channel messages. All Pull Directory messages are formatted as

described below starting with the following common 8-byte header:



Ver: Version of the Pull Directory protocol as an unsigned integer. Version zero is specified in this document.

Type: The Pull Directory message type as follows:

Type	Section	Name
0	3.2.1	Query
1	3.2.2	Response
2	3.1.4	Update
3	3.1.5	Acknowledge
4-15	-	Reserved

Flags: Four flag bits whose meaning depends on the Pull Directory message Type. Flags whose meaning is not specified are reserved, MUST be sent as zero, and ignored on receipt.

Count: Most Pull Directory message types specified herein have zero or more occurrences of a Record as part of the type specific payload. The Count field is the number of occurrences of that Record as an unsigned integer. For Pull Directory messages not structured with such occurrences, this field MUST be sent as zero and ignored on receipt.

Err, SubErr: The error and suberror fields are only used in messages that are in the nature of replies or acknowledgements. In messages that are requests or updates, these fields MUST be sent as zero and ignored on receipt. The meaning of values in the Err field depends on the Pull Directory message Type but in all cases the value zero means no error. The meaning of values in the SubErr field depends on both the message Type and on the value of the Err field but in all cases, a zero SubErr field is allowed and provides no additional information beyond the value of the Err field.

Sequence Number: An opaque 32-bit quantity set by the TRILL switch sending a request or other unsolicited message and returned in any reply or acknowledgement. It is used to match up responses

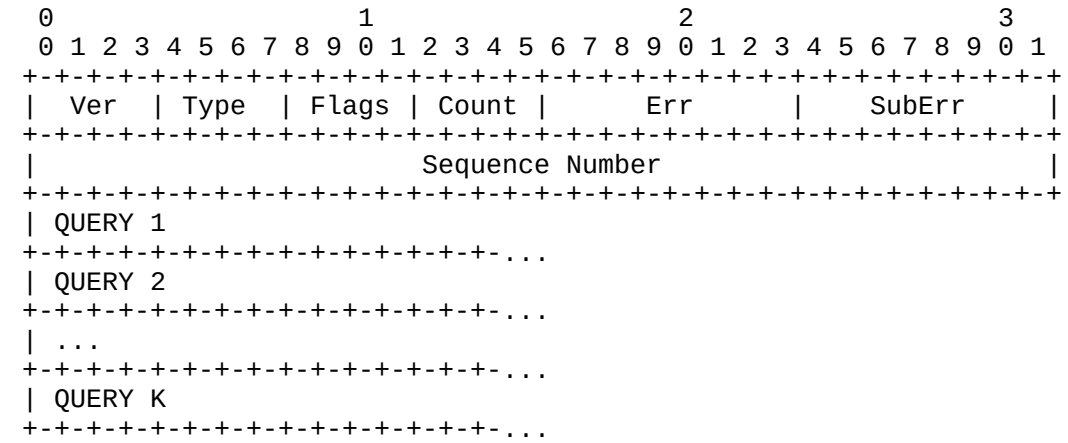
with the message to which they respond.

Type Specific Payload: Format depends on the Pull Directory message Type.

3.2 Pull Directory Query and Response Messages

3.2.1 Pull Directory Query Message Format

A Pull Directory Query message is sent as the Channel Protocol specific content of an RBridge Channel message [Channel] TRILL Data packet or as a native RBridge Channel data frame (see Section 3.4). The Data Label of the packet is the Data Label in which the query is being made. The priority of the channel message is a mapping of the priority of the frame being ingressed that caused the query with the default mapping depending, per Data Label, on the strategy (see Section 6) or a configured priority for generated queries. The Channel Protocol specific data is formatted as a header and a sequence of zero or more QUERY Records as follows:



Ver, Sequence Number: See 3.1.

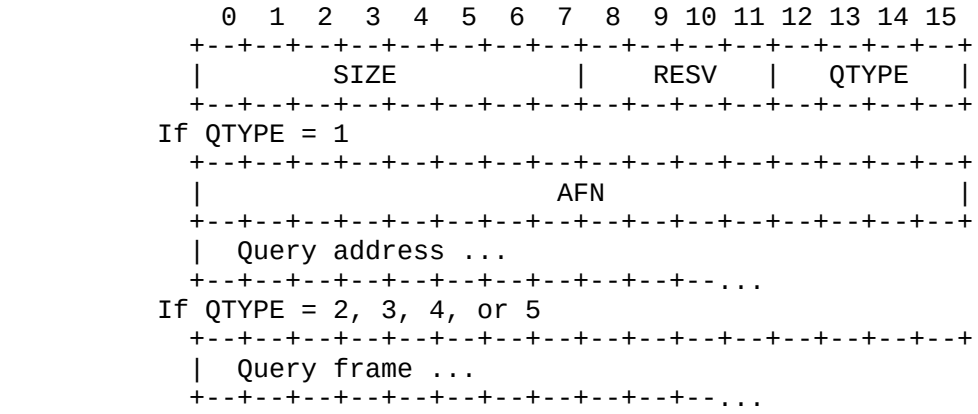
Type: 1 for Query. Queries received by an TRILL switch that is not a Pull Directory result in an error response (see Section 3.5) unless inhibited by rate limiting.

Flags, Err, and SubErr: MUST be sent as zero and ignored on receipt.

Count: Number of QUERY Records present. A Query message Count of

zero is explicitly allowed, for the purpose of pinging a Pull Directory server to see if it is responding. On receipt of such an empty Query message, a Response message that also has a Count of zero is sent unless inhibited by rate limiting.

QUERY: Each QUERY Record within a Pull Directory Query message is formatted as follows:



SIZE: Size of the QUERY record in bytes as an unsigned integer starting after the SIZE field and following byte. Thus the minimum legal value is 2. A value of SIZE less than 2 indicates a malformed QUERY record. The QUERY record with the illegal SIZE value and any subsequent QUERY records MUST be ignored and the entire Query message MAY be ignored.

RESV: A block of reserved bits. MUST be sent as zero and ignored on receipt.

QTYPE: There are several types of QUERY Records currently defined in two classes as follows: (1) a QUERY Record that provides an explicit address and asks for all addresses for the interface specified by the query address and (2) a QUERY Record that includes a frame. The fields of each are specified below. Values of QTYPE are as follows:

QTYPE	Description
0	reserved
1	address query
2	ARP query frame
3	ND query frame
4	RARP query frame
5	Unknown unicast MAC query frame
6-14	assignable by IETF Review
15	reserved

AFN: Address Family Number of the query address.

Address Query: The query is asking for any other addresses, and the nickname of the TRILL switch from which they are reachable, that correspond to the same interface, within the data label of the query. Typically that would be either (1) a MAC address with the querying TRILL switch primarily interested in the TRILL switch by which that MAC address is reachable, or (2) an IP address with the querying TRILL switch interested in the corresponding MAC address and the TRILL switch by which that MAC address is reachable. But it could be some other address type.

Query Frame: Where a QUERY Record is the result of an ARP, ND, RARP, or unknown unicast MAC destination address, the ingress TRILL switch MAY send the frame to a Pull Directory Server if the frame is small enough that the resulting Query message fits into a TRILL Data packet within the campus MTU.

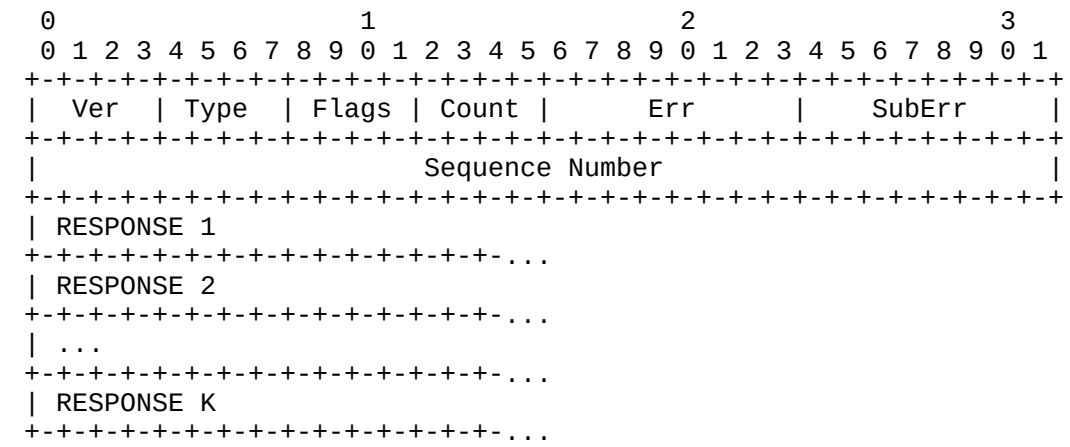
If no response is received to a Pull Directory Query message within a timeout configurable in milliseconds that defaults to 200, the Query message should be re-transmitted with the same Sequence Number up to a configurable number of times that defaults to three. If there are multiple QUERY Records in a Query message, responses can be received to various subsets of these QUERY Records before the timeout. In that case, the remaining unanswered QUERY Records should be re-sent in a new Query message with a new sequence number. If a TRILL switch is not capable of handling partial responses to queries with multiple QUERY Records, it MUST NOT sent a Request message with more than one QUERY Record in it.

See Section 3.5 for a discussion of how Query message errors are handled.

3.2.2 Pull Directory Response Format

Pull Directory Response messages are sent as the Channel Protocol specific content of an RBridge Channel message [Channel] TRILL Data packet or as a native RBridge Channel data frame (see Section 3.4). Responses are sent with the same Data Label and priority as the Query message to which they correspond except that the Response message priority is limited to be not more than a configured value. This priority limit is configurable at per TRILL switch and defaults to priority 6. Pull Directory Response messages SHOULD NOT be sent with priority 7 as that priority SHOULD be reserved for messages critical to network connectivity.

The RBridge Channel protocol specific data format is as follows:



Ver, Sequence Number: As specified in Section 3.1.

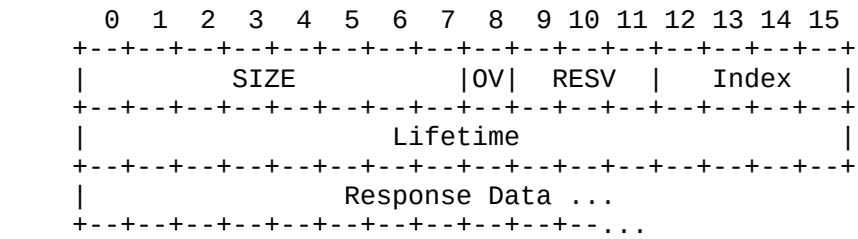
Type: 2 = Response.

Flags: MUST be sent as zero and ignored on receipt.

Count: Count is the number of RESPONSE Records present in the Response message.

Err, SubErr: A two part error code. Zero unless there was an error in the Query message, for which case see Section 3.5.

RESPONSE: Each RESPONSE record within a Pull Directory Response message is formatted as follows:



SIZE: Size of the RESPONSE Record in bytes starting after the SIZE field and following byte. Thus the minimum value of SIZE is 2. If SIZE is less than 2, that RESPONSE Record and all subsequent RESPONSE Records in the Response message MUST be ignored and the entire Response message MAY be ignored.

OV: The overflow flag. Indicates, as described below, that there was too much Response Data to include in one Response

message.

RESV: Four reserved bits that MUST be sent as zero and ignored on receipt.

Index: The relative index of the QUERY Record in the Query message to which this RESPONSE Record corresponds. The index will always be one for Query messages containing a single QUERY Record. If the Index is larger than the Count was in the corresponding Query, that RESPONSE Record MUST be ignored and subsequent RESPONSE Records or the entire Response message MAY be ignored.

Lifetime: The length of time for which the response should be considered valid in units of 200 milliseconds except that the values zero and 2**16-1 are special. If zero, the response can only be used for the particular query from which it resulted and MUST NOT be cached. If 2**16-1, the response MAY be kept indefinitely but not after the Pull Directory server goes down or becomes unreachable. The maximum definite time that can be expressed is a little over 3.6 hours.

Response Data: There are various types of RESPONSE Records.
- If the Err field is non-zero, then the Response Data is a copy of the corresponding QUERY Record data, that is, either an AFN followed by an address or a query frame. See Section 3.5 for additional information on errors.
- If the Err field is zero and the corresponding QUERY Record was an address query, then the Response Data is the contents of an Interface Addresses APPsub-TLV [IA]. The maximum size of such contents is 253 bytes in the case when SIZE is 255.
- If the Err field is zero and the corresponding QUERY Record was a frame query, then the Response data consists of the response frame for ARP, ND, or RARP and a copy of the frame for unknown unicast destination MAC.

Multiple RESPONSE Records can appear in a Response message with the same index if the answer to a QUERY Record consists of multiple Interface Address APPsub-TLV contents. This would be necessary if, for example, a MAC address within a Data Label appears to be reachable by multiple TRILL switches. However, all RESPONSE Records to any particular QUERY Record MUST occur in the same Response message. If a Pull Directory holds more mappings for a queried address than will fit into one Response message, it selects which to include by some method outside the scope of this document and sets the overflow flag (OV) in all of the RESPONSE Records responding to that query address.

See Section 3.5 for a discussion of how errors are handled.

3.3 Cache Consistency

A Pull Directory MUST take action to minimize the amount of time that a TRILL switch will continue to use stale information from that Pull Directory by sending Update messages.

A Pull Directory server MUST maintain one of the following three sets of records, in order of increasing specificity. Retaining more specific records, such as that given in item 3 below, minimizes Spontaneous Update messages sent to update pull client TRILL switch caches but increases the record keeping burden on the Pull Directory server. Retaining less specific records, such as that given in item 1, will generally increase the volume and overhead due to Spontaneous Update messages and due to unnecessarily invalidating cached information, but will still maintain consistency and will reduce the record keeping burden on the Pull Directory server. In all cases, there may still be brief periods of time when directory information has changed but cached information a pull clients has not yet been updated or expunged.

1. An overall record per Data Label of when the last positive response data sent will expire at some requester and when the last negative response will expire at some requester, assuming those responders cached the response.

2. For each unit of data (IA APPsub-TLV Address Set [IA]) held by the server and each address about which ‘a negative response was sent, when the last response sent with that positive response data or negative response will expire at a requester, assuming the requester cached the response.

3. For each unit of data held by the server (IA APPsub-TLV Address Set [IA]) and each address about which a negative response was sent, a list of TRILL switches that were sent that data as a positive response or sent a negative response for the address, and the expected time to expiration for that data or address at each such TRILL switch, assuming the requester cached the response.

A Pull Directory server may have a limit as to how many TRILL switches for which it can maintain expiry information by method 3 above or how many data units or addresses it can maintain expiry information for by method 2. If such limits are exceeded, it MUST transition to a lower numbered strategy but, in all cases, MUST support, at a minimum, method 1.

When data at a Pull Directory changes or is deleted or data is added and there may be unexpired stale information at a requesting TRILL switch, the Pull Directory MUST send an Update message as discussed below. The sending of such an Update message MAY be delayed by a configurable number of milliseconds that default to 50 milliseconds to await other possible changes that could be included in the same Update.

If method 1, the most crude method, is being followed, then when any Pull Directory information in a Data Label is changed or deleted and there are outstanding cached positive data response(s), an all-addresses flush positive Update message is flooded within that Data Label as an RBridge Channel message with an Inner.MacDA of All-Egress-RBridges. And if data is added and there are outstanding cached negative responses, an all-addresses flush negative message is similarly flooded. "All-addresses" is indicated by the Count field being zero in an Update message. On receiving an all-addresses flooded flush positive Update from a Pull Directory server it has used, indicated by the F and P bits being one and the Count being zero, a TRILL switch discards all cached data responses it has for that Data Label. Similarly, on receiving an all addresses flush negative Update, indicated by the F and N bits being one and the Count being zero, it discards all cached negative replies for that Data Label. A combined flush positive and negative can be flooded by having all of the F, P, and N bits set to one resulting in the discard of all positive and negative cached information for the Data Label.

If method 2 is being followed, then a TRILL switch floods address specific positive Update messages when data that might be cached by a querying TRILL switch is changed or deleted and floods address specific negative Update messages when such information is added to. Such messages are similar to the method 1 flooded flush Update messages and are also sent as RBridge Channel messages with an Inner.MacDA of All-Egress-RBridges. However the Count field will be non-zero and either the P or N bit, but not both, will be one. On receiving such as address specific unsolicited update, if it is positive the addresses in the RESPONSE records in the unsolicited response are compared to the addresses about which the receiving TRILL switch is holding cached positive information from that server and, if they match, the cached information is updated. On receiving an address specific unsolicited update negative message, the addresses in the RESPONSE records in the unsolicited update are compared to the addresses about which the receiving TRILL switch is holding cached negative information from that server and, if they match, the cached negative information is updated.

If method 3 is being followed, the same sort of unsolicited update messages are sent as with method 2 above except they are not normally flooded but unicast only to the specific TRILL switches the directory

server believes may be holding the cached positive or negative information that needs updating. However, a Pull Directory server MAY flood the unsolicited update under method 3, for example if it determines that a sufficiently large fraction of the TRILL switches in some Data label are requesters that need to be updated.

A Pull Directory server tracking cached information with method 3 MUST NOT clear the indication that it needs update cached information at a querying TRILL switch until it has sent an Update message and received a corresponding Acknowledge message or it has sent a configurable number of updates at a configurable interval which default to 3 updates 200 milliseconds apart.

A Pull Directory server tracking cached information with methods 2 or 1 SHOULD NOT clear the indication that it needs to update cached information until it has sent an Update message and received a corresponding Acknowledge message from all of its ESADI neighbors or it has sent a configurable number of updates at a configurable interval that defaults to 3 updates 200 milliseconds apart.

3.3.1 Update Message Format

An Update message is formatted as a Response message except that the Type field in the message header is a different value.

Update messages are initiated by a Pull Directory server. The Sequence number space used is controlled by the originating Pull Directory server and different from Sequence number space used in a Query and the corresponding Response that are controlled by the querying TRILL switch.

The Flags field of the message header for an Update message is as follows:

F: The Flood bit. If zero, the response is to be unicast . If F=1, it is multicast to All-Egress-RBridges.

P, N: Flags used to indicate positive or negative Update messages. P=1 indicates positive. N=1 indicates negative. Both may be 1 for a flooded all addresses Update.

R: Reserved. MUST be sent as zero and ignored on receipt

3.3.2 Acknowledge Message Format

An Acknowledge message is sent in response to an Update to confirm receipt or indicate an error unless response is inhibited by rate limiting. It is also formatted as a Response message.

If there are no errors in the processing of an Update message, the message is essentially echoed back with the Type changed to Acknowledge.

If there was an overall or header error in an Update message, it is echoed back as an Acknowledge message with the Err and SubErr fields set appropriately (see Section 3.5).

If there is a RESPONSE Record level error in an Update message, one or more Acknowledge messages may be returns as indicated in Section 3.5.

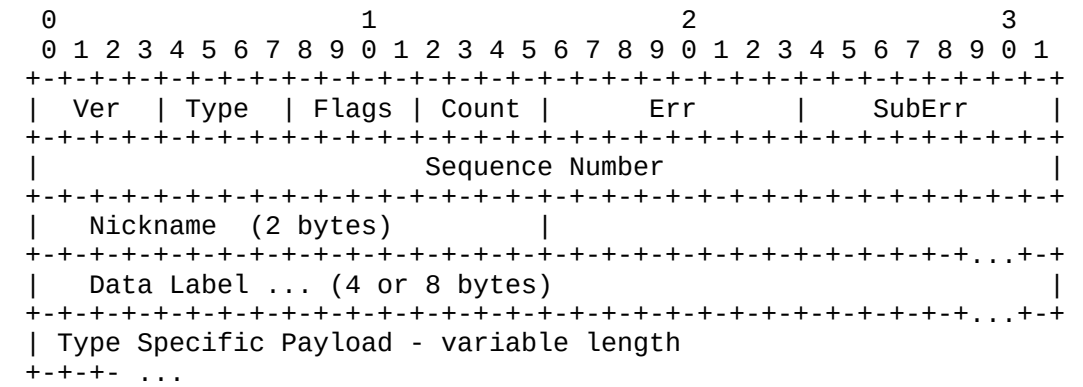
3.4 Pull Directory Hosted on an End Station

Optionally, a Pull Directory actually hosted on an end station MAY be supported. In that case, a TRILL switch must proxy for the end station and advertise itself as a Pull Directory server.

When the proxy TRILL switch receives a Query message, it modifies the inter-RBridge Channel message received into a native RBridge Channel message and forwards it to that end station. Later, when it receives one or more responses from that end station by native RBridge Channel messages, it modifies them into inter-RBridge Channel messages and forwards them to the source TRILL switch of the original Query message. Similarly, an Update from the end station is forwarded to client TRILL switches and acknowledgements from those TRILL switches are returned to the end station by the proxy. Because native RBridge Channel messages have no TRILL Header and are addressed by MAC address, as opposed to inter-RBridge Channel messages that are TRILL Data packets and are addressed by nickname, nickname information must be added to the native RBridge Channel version of Pull Directory messages.

The native Pull Directory RBridge Channel messages use the same Channel protocol number as do the inter-RBridge Pull Directory RBridge Channel messages. The native messages SHOULD be sent with an Outer.VLAN tag which gives the priority of each message which is the priority of the original inter-RBridge request packet. The Outer.VLAN ID used is the Designated VLAN on the link to the end station. Since there is no TRILL Header or inner Data Label for native RBridge Chanel messages, that information is added to the header.

The native RBridge Channel message protocol dependent data Pull Directory message is the same as for inter-RBridge Channel messages except that the 8-byte header described in Section 3.1 is expanded to 14 or 18 bytes as follows:



Fields not described below are as in Section 3.1.

Data Label: The Data Label that normally appear right after the Inner.MacSA of the an RBridge Channel Pull Directory message appears here in the native RBridge Channel message version. This might appear in a Query message, to be reflected in a Response message, or it might appear in an Update message, to be reflected in an Acknowledge message.

Nickname: The nickname of the TRILL switch that is communicating with the end station Pull Directory. Usually this is a remote TRILL switch but it could be the TRILL switch to which the end station is attached. The proxy copies this from the ingress nickname when mapping a Query or Acknowledge message to native form. It also takes this from a native Response or Update to be used as the egress of the inter-RBridge form on the message unless it is a flooded Update in which case a distribution tree is used.

3.5 Pull Directory Message Errors

A non-zero Err field in the Pull Directory message header indicates an error message.

If there is an error that applies to an entire Query message or its header, as indicated by the range of the value of the Err field, then the QUERY records in the request are just echoed back in the RESPONSE records of the Response message but expanded with a zero Lifetime and the insertion of the Index field. If there is an error that applies

to an entire Update message or its header, then the RESPONSE records in the update, if any, are echoed back in the Acknowledge message.

If errors occur at the QUERY Record level for a Query message, they MUST be reported in a Response message separate from the results of any successful non-erroneous QUERY Records. If multiple QUERY Records in a Query message have different errors, they MUST be reported in separate Response messages. If multiple QUERY Records in a Query message have the same error, this error response MAY be reported in one Response message. In an error Response message, the QUERY Record or records being responded to appear, expanded by the Lifetime for which the server thinks the error might persist and with their Index inserted, as the RESPONSE record or records.

If errors occur at the RESPONSE Record level for an Update message, they MUST be reported in a Acknowledge message separate from the acknowledgement of any non-erroneous RESPONSE Records. If multiple RESPONSE Records in an Update have different errors, they MUST be reported in separate Acknowledge messages. If multiple RESPONSE Records in an Update message have the same error, this error response MAY be reported in one Acknowledge message. In an error Acknowledge message, the RESPONSE Record or records being responded to appear, expanded by the time for which the server thinks the error might persist and with their Index inserted, as a RESPONSE Record or records.

ERR values 1 through 127 are available for encoding Request or Update message level errors. ERR values 128 through 254 are available for encoding QUERY or RESPONSE Record level errors. The SubErr field is available for providing more detail on errors. The meaning of a SubErr field value depends on the value of the Err field.

Err	Meaning
0	(no error)
1	Unknown or reserved Query message field value
2	Request data too short
3	Unknown or reserved Update message field value
4	Update data too short
5-127	(Available for allocation by IETF Review)
128	Unknown or reserved QUERY Record field value
129	Address not found
130	Unknown or reserved RESPONSE Record field value
131-254	(Available for allocation by IETF Review)
255	Reserved

The following sub-errors are specified under error code 1 and 3:

SubErr	Field with Error
0	Unspecified
1	Unknown V field value
2	Reserved T field value
3	Zero sequence number in request
4-254	(Available for allocation by Expert Review)
255	Reserved

The following sub-errors are specified under error code 128 and 130:

SubErr	Field with Error
0	Unspecified
1	Unknown AFN field value
2	Unknown or Reserved TYPE field value
3	Invalid or inconsistent SIZE field value
4-254	(Available for allocation by Expert Review)
255	Reserved

More TBD

3.6 Additional Pull Details

If a TRILL switch notices that a Pull Directory server is no longer data reachable [RFCclear], it MUST promptly discard all pull responses it is retaining from that server as it can no longer receive cache consistency update messages from the server.

Because a Pull Directory server may need to advertise interest in Data Labels even though it does not want to received end station data in those Data Labels, the No Data (NOD) flag bit is provided as specified in Section 8.3. For example, an RBridge hosting a Pull Directory may be a secondary directory that wants to receive its data from a primary Push Directory server but have no interest in receiving multicast traffic from end stations.

4. Events That May Cause Directory Use

A TRILL switch can consult Directory information whenever it wants, by (1) searching through information that has been retained after being pushed to it or pulled by it or (2) by requesting information from a Pull Directory. However, the following are expected to be the most common circumstances leading to directory information use. All of these are cases of ingressing (or originating) a native frame.

ARP requests and replies normally have the broadcast address in their MAC destination address and are normally treated the same way as any broadcast Ethernet frame. A directory assisted RBridge MUST intercept ARP broadcast, ND multicast, and unknown unicast destination MAC address native frames. It SHOULD also intercept RARP and, if complete directory information is available, forged source MAC frames.

Support for each of the cases below is separately optional.

4.1 Forged Native Frame Ingress

End stations can forge the source MAC and/or IP address in a native frame that an edge TRILL switch receives for ingress in some particular Data Label. If there is complete Directory information as to what end stations should be reachable by an egress TRILL switch, frames with forged source addresses SHOULD be discarded. If such frames are discarded, then none of the special processing in the remaining subsection of this Section 2 occur and MAC address learning (see [RFC6325] Section 4.8) SHOULD NOT occur. ("SHOULD NOT" is chosen because it is harmless in cases where it has no effect. For example, if complete directory information is available and such directory information is treated as having a higher confidence that MAC addresses learned from the data plane.)

If directory information includes the TRILL switch a port by which a MAC and/or IP address is reachable, that may also be tested on ingress so that an end station on one TRILL switch port cannot forge a source MAC or IP address that should not be reachable by that port even if it is reachable by that TRILL switch.

4.2 Unknown Destination MAC

Ingressing a native frame with an unknown unicast destination MAC:

The mapping from the destination MAC and Data Label to the egress TRILL switch from which it is reachable is needed to ingress the frame as unicast. If the egress TRILL switch is unknown, the frame

must be either dropped or ingressed as a multi-destination frame which is flooded to all edge TRILL switches for its Data Label resulting in increased link utilization compared with unicast routing. Depending on the configuration of the TRILL switch ingressing the native frame (see Section 6), directory information can be used for the { destination MAC, Data Label } to egress TRILL switch nickname mapping and destination MACs for which such direction information is not available MAY be discarded.

4.3 Address Resolution Protocol (ARP)

Ingressing an ARP [RFC826]:

ARP is a flexible protocol detected by its Ethertype of 0x0806. It is commonly used on a link to (1) query for the MAC address corresponding to an IPv4 address, (2) test if an IPv4 address is in use, or (3) to announce a change in any of IPv4 address, MAC address, and/or point of attachment.

The logically important elements in an ARP are (1) the specification of a "protocol" and a "hardware" address type, (2) an operation code that can be Request or Reply, and (3) fields for the protocol and hardware address of the sender and the target (destination) node.

Examining the three types of ARP use:

1. General ARP Request / Response

   This is a request for the destination "hardware" address corresponding to the destination "protocol" address; however, if the source and destination protocol addresses are equal, it should be handled as in type 2 below. A general ARP is handled by doing a directory lookup on the destination "protocol" address provided in hops of finding a mapping to the desired "hardware" address. If such information is obtain from a directory, a response can be synthesized.

2. Address Probe ARP Query

   An address probe ARP is used to determine if an IPv4 address is in use [RFC5227]. It can be identified by the source "protocol" (IPv4) address field being zero. The destination "protocol" address field is the IPv4 address being tested. If some host believes it has that destination IPv4 address, it would respond to the ARP query, which indicates that the address is in use. Address probe ARPs can be handled in the same way as General ARP queries above.

3. Gratuitous ARP

A gratuitous ARP is an unsolicited ARP message, usually a response but sometimes a query, used by a host to announce a new IPv4 address, new MAC address, and/or new point of network attachment. Such ARPs are identifiable because the sender and destination "protocol" address fields have the same value. Thus, under normal circumstances, there really isn't any separate destination host to generate a response. If complete Push Directory information is being used with the Notify flag set in the IA APPsub-TLVs being pushed [IA] by all the TRILL switches in the Data Label, then gratuitous ARPs SHOULD be discarded rather the ingressed. Otherwise, they are either ingressed and flooded or discarded depending on local policy.

4.4 IPv6 Neighbor Discovery (ND)

Ingressing an IPv6 ND [RFC4861]:
TBD

Secure Neighbor Discovery messages [RFC3971] will, in general, have to be sent to the neighbor intended so that neighbor can sign the answer; however, directory information can be used to unicast a Secure Neighbor Discovery packet rather than multicasting it.

4.5 Reverse Address Resolution Protocol (RARP)

Ingressing a RARP [RFC903]:
RARP uses the same packet format as ARP but a different Ethertype (0x8035) and opcode values. Its use is similar to the General ARP Request/Response as described above. The difference is that it is intended to query for the destination "protocol" address corresponding to the destination "hardware" address provided. It is handled by doing a directory lookup on the destination "hardware" address provided in hops of finding a mapping to the desired "protocol" address. For example, looking up a MAC address to find the corresponding IP address.

5. Layer 3 Address Learning

TRILL switches MAY learn IP addresses in a manner similar to that in which they learn MAC addresses. On ingress of a native IP frame, they can learn the { IP address, MAC address, Data Label, input port } set and on the egress of a native IP frame, they can learn the { IP address, MAC address, Data Label, remote RBridge } information plus the nickname of the RBridge that ingressed the frame.

This locally learned information is retained and times out in a similar manner to MAC address learning specified in [RFC6325]. By default, it has the same Confidence as locally learned MAC reachability information.

Such learned Layer 3 address information MAY be disseminated with ESDADI [RFCesadi] using the IA APPsub-TLV [IA]. It can also be used as, in effect, local directory information to assist in locally responding to ARP/ND packets as discussed in Section 4.

6. Directory Use Strategies and Push-Pull Hybrids

For some edge nodes that have a great number of Data Labels enabled, managing the MAC and Data Label <-> Edge RBridge mapping for hosts under all those Data Labels can be a challenge. This is especially true for Data Center gateway nodes, which need to communicate with a majority of Data Labels, if not all.

For those edge TRILL switch nodes, a hybrid model should be considered. That is the Push Model is used for some Data Labels, and the Pull Model is used for other Data Labels. It is the network operator's decision by configuration as to which Data Labels' mapping entries are pushed down from directories and which Data Labels' mapping entries are pulled.

For example, assume a data center where hosts in specific Data Labels, say VLANs 1 through 100, communicate regularly with external peers. Probably, the mapping entries for those 100 VLANs should be pushed down to the data center gateway routers. For hosts in other Data Labels which only communicate with external peers occasionally for management interface, the mapping entries for those VLANs should be pulled down from directory when the need comes up.

The mechanisms described above for Push and Pull Directory services make it easy to use Push for some Data Labels and Pull for others. In fact, different TRILL switches can even be configured so that some use Push Directory services and some use Pull Directory services for the same Data Label if both Push and Pull Directory services are available for that Data Label. And there can be Data Labels for which directory services are not used at all.

For Data Labels in which a hybrid push/pull approach is being taken, it would make sense to use push for address information of hosts that frequently communicate with many other hosts in the Data Label, such as a file or DNS server. Pull could then be used for hosts that communicate with few other hosts, perhaps such as hosts being used as compute engines.

6.1 Strategy Configuration

Each TRILL switch that has the ability to use directory assistance has, for each Data Label X in which it is might ingress native frames, one of four major modes:

0. No directory use: The TRILL switch does not subscribe to Push Directory data or make Pull Directory requests for Data Label X and directory data is not consulted on ingressed frames in Data Label X that might have used directory data. This includes ARP,

ND, RARP, and unknown MAC destination addresses, which are flooded as appropriate.

1. Use Push only: The TRILL switch subscribes to Push Directory data for Data Label X.

2. Use Pull only: When the TRILL switch ingresses a frame in Data Label X that can use Directory information, if it has cached information for the address it uses it. If it does not have either cached positive or negative information for the address, it sends a Pull Directory query.

3. Use Push and Pull: The TRILL switch subscribes to Push Directory data for Data Label X. When it ingresses a frame in Data Label X that can use Directory information and it does not find that information in its link state database of Push Directory information, it makes a Pull Directory query.

The above major Directory use mode is per Data Label. In addition, there is a per Data Label per priority minor mode as listed below that indicates what should be done if Directory Data is not available for the ingressed frame. In all cases, if you are holding Push Directory or Pull Directory information to handle the frame given the major mode, the directory information is simply used and, in that instance, the minor mode does not matter.

A. Flood immediate: Flood the frame immediately (even if you are also sending a Pull Directory) request.

B. Flood: Flood the frame immediately unless you are going to do a Pull Directory request, in which case you wait for the response or for the request to time out after retries and flood the frame if the request times out.

C. Discard if complete or Flood immediate: If you have complete Push Directory information and the address is not in that information, discard the frame. If you do not have complete Push Directory information, the same as A above.

D. Discard if complete or Flood: If you have complete Push Directory information and the address is not in that information, discard the frame. If you do not have complete Push Directory information, the same as B above.

In addition, the query message priority for Pull Directory requests sent can be configured on a per Data Label, per ingressed frame priority basis. The default mappings are as follows where Ingress Priority is the priority of the native frame that provoked the Pull Directory query:

Ingress Priority	If Flood Immediate	If Flood Delayed
7	5	6
6	5	6
5	4	5
4	3	4
3	2	3
2	0	2
0	1	0
1	1	1

Priority 7 is normally only used for urgent messages critical to adjacency and so is avoided by default for directory traffic. Unsolicited updates are sent with a priority that is configured per Data Label that defaults to priority 5.

7. Security Considerations

Incorrect directory information can result in a variety of security threats including the following:

- Incorrect directory mappings can result in data being delivered to the wrong end stations, or set of end stations in the case of multi-destination packets, violation security policy.

- Missing or incorrect directory data can result in denial of service due to sending data packets to black holes or discarding data on ingress due to incorrect information that their destinations are not reachable.

Push Directory data is distributed through ESADI-LSPs [RFCesadi] that can be authenticated with the same mechanisms as IS-IS LSPs. See [RFC5304] [RFC5310] and the Security Considerations section of [RFCesadi].

Pull Directory queries and responses are transmitted as RBridge-to-RBridge or native RBridge Channel messages. Such messages can be secured as specified in [ChannelTunnel].

For general TRILL security considerations, see [RFC6325].

8. IANA Considerations

This section gives IANA allocation and registry considerations.

8.1 ESADI-Parameter Data Extensions

IANA is requested to allocate two ESADI-Parameter TRILL APPsub-TLV flag bits for "Push Directory" (PSH) and "Complete Push" (COP) and to create a sub-registry in the TRILL Parameters Registry as follows:

Sub-Registry: ESADI-Parameter APPsub-TLV Flag Bits

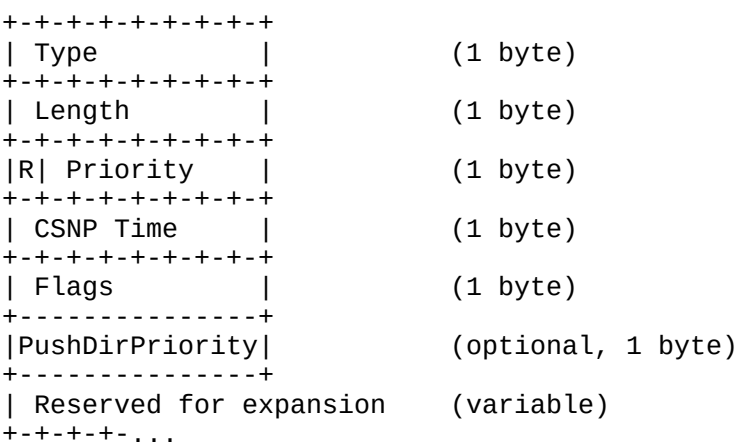
Registration Procedures: Expert Review

References: [RFCesadi] [This document]

Bit	Mnemonic	Description	Reference
0	UN	Supports Unicast ESADI	ESDADI [RFCesadi]
1	PSH	Push Directory Server	This document
2	COP	Complete Push	This document
3-7	-	available for allocation	

The COP bit is ignored if the PSH bit is zero.

In addition, the ESADI-Parameter APPsub-TLV is optionally extended, as provided in its original specification in ESDADI [RFCesadi], by one byte as show below:



The meanings of all the fields are as specified in ESDADI [RFCesadi] except that the added PushDirPriority is the priority of the advertising ESADI instance to be a Push Directory as described in

Section 2.3. If the PushDirPriority field is not present (Length = 3) it is treated as if it were 0x40. 0x40 is also the value used and placed here by an TRILL switch whose priority to be a Push Directory has not been configured.

8.2 RBridge Channel Protocol Number

IANA is requested to allocate a new RBridge Channel protocol number for "Pull Directory Services" from the range allocable by Standards Action and update the subregistry of such protocol number in the TRILL Parameters Registry referencing this document.

8.3 The Pull Directory (PUL) and No Data (NOD) Bits

IANA is requested to allocate two currently reserved bits in the Interested VLANs field of the Interested VLANs sub-TLV (suggested bits 18 and 19) and the Interested Labels field of the Interested Labels sub-TLV (suggested bits 6 and 7) [RFC6326bis] to indicate Pull Directory server (PUL) and No Data (NOD) respectively. These bits are to be added, with this document as reference, to the "Interested VLANs Flag Bits" and "Interested Labels Flag Bits" subregistries created by [RFCesadi].

In the TRILL base protocol [RFC6325] as extended for FGL [rfcFGL], the mere presence of an Interested VLANs or Interested Labels sub-TLVs in the LSP of a TRILL switch indicates connection to end stations in the VLAN(s) or FGL(s) listed and thus a desire to receive multi-destination traffic in those Data Labels. But, with Push and Pull Directories, advertising that you are a directory server requires using these sub-TLVs to indicate the Data Label(s) you are serving. If such a directory server does not wish to received multi-destination TRILL Data packets for the Data Labels it lists in one of these sub-TLVs, it sets the "No Data" (NOD) bit to one. This means that data on a distribution tree may be pruned so as not to reach the "No Data" TRILL switch as long as there are no TRILL switches interested in the Data that are beyond the "No Data" TRILL switch on a distribution tree. The NOD bit is backwards compatible as TRILL switches ignorant of it will simply not prune when they could, which is safe although it may cause increased link utilization.

An example of a TRILL switch serving as a directory that would not want multi-destination traffic in some Data Labels might be a TRILL switch that does not offer end station service for any of the Data Labels for which it is serving as a directory and is either a Pull Directory and/or a Push Directory for which all of the ESADI traffic can be handled by unicast ESDADI [RFCesadi].

Acknowledgments

The contributions of the following persons are gratefully acknowledged:

TBD

The document was prepared in raw nroff. All macros used were defined within the source file.

Normative References

[RFC826] - Plummer, D., "An Ethernet Address Resolution Protocol", RFC 826, November 1982.

[RFC903] - Finlayson, R., Mann, T., Mogul, J., and M. Theimer, "A Reverse Address Resolution Protocol", STD 38, RFC 903, June 1984

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

[RFC3971] - Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4861] - Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, October 2008.

[RFC5310] - Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

[RFC6165] - Banerjee, A. and D. Ward, "Extensions to IS-IS for Layer-2 Systems", RFC 6165, April 2011.

[RFC6325] - Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

[RFC7042] - Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 7042, October 2013.

[RFC6326bis] - Eastlake, D., Banerjee, A., Dutt, D., Perlman, R., and A. Ghanwani, "TRILL Use of IS-IS", draft-ietf-isis-rfc6326bis, work in progress.

[RFCclear] - Eastlake, D., M. Zhang, A. Ghanwani, V. Manral, A. Banerjee, draft-ietf-trill-clear-correct-06.txt, in RFC Editor’s queue.

[Channel] - D. Eastlake, V. Manral, Y. Li, S. Aldrin, D. Ward, "TRILL: RBridge Channel Support", draft-ietf-trill-rbridge-channel-08.txt, in RFC Editor’s queue.

[RFCfgl] - D. Eastlake, M. Zhang, P. Agarwal, R. Perlman, D. Dutt,

"TRILL: Fine-Grained Labeling", draft-ietf-trill-fine-labeling-07.txt, in RFC Editor's queue.

[RFCesadi] - Zhai, H., F. Hu, R. Perlman, D. Eastlake, O. Stokes, "TRILL (Transparent Interconnection of Lots of Links): The ESADI (End Station Address Distribution Information) Protocol", draft-ietf-trill-esadi, work in progress.

[IA] - Eastlake, D., L. Yizhou, R. Perlman, "TRILL: Interface Addresses APPsub-TLV", draft-eastlake-trill-ia-appsubtlv, work in progress.

Informational References

[RFC5227] - Cheshire, S., "IPv4 Address Conflict Detection", RFC 5227, July 2008.

[RFC7067] - Dunbar, L., Eastlake 3rd, D., Perlman, R., and I. Gashinsky, "Directory Assistance Problem and High-Level Design Proposal", RFC 7067, November 2013.

[ChannelTunnel] - D. Eastlake, Y. Li, "TRILL: RBridge Channel Tunnel Protocol", draft-eastlake-trill-channel-tunnel, work in progress.

[ARP reduction] - Shah, et. al., "ARP Broadcast Reduction for Large Data Centers", Oct 2010.

Authors' Addresses

Linda Dunbar
Huawei Technologies
5430 Legacy Drive, Suite #175
Plano, TX 75024, USA

Phone: +1-469-277-5840
Email: ldunbar@huawei.com

Donald Eastlake
Huawei Technologies
155 Beaver Street
Milford, MA 01757 USA

Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

Radia Perlman
Intel Labs
2200 Mission College Blvd.
Santa Clara, CA 95054-1549 USA

Phone: +1-408-765-8080
Email: Radia@alum.mit.edu

Igor Gashinsky
Yahoo
45 West 18th Street 6th floor
New York, NY 10011

Email: igor@yahoo-inc.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012 China

Phone: +86-25-56622310
Email: liyizhou@huawei.com

Copyright, Disclaimer, and Additional IPR Provisions

INTERNET-DRAFT
Intended status: Proposed Standard

Donald Eastlake
Yizhou Li
Huawei
Radia Perlman
Intel

Expires: December 1, 2014
June 2, 2014

TRILL: Interface Addresses APPsub-TLV
<draft-ietf-trill-ia-appsubtlv-01.txt>

Abstract

This document specifies a TRILL (Transparent Interconnection of Lots of Links) IS-IS application sub-TLV that enables the reporting by a TRILL switch of sets of addresses such that all of the addresses in each set designate the same interface (port) and the reporting for such a set of the TRILL switch by which it is reachable. For example, a 48-bit MAC (Media Access Control) address, IPv4 address, and IPv6 address can be reported as all corresponding to the same interface reachable by a particular TRILL switch. Such information could be used in some cases to synthesize responses to or by-pass the need for the Address Resolution Protocol (ARP), the IPv6 Neighbor Discovery (ND) protocol, or the flooding of unknown MAC addresses.

Table of Contents

1. Introduction

This document specifies a TRILL (Transparent Interconnection of Lots of Links) [RFC6325] IS-IS application sub-TLV (APPsub-TLV [RFC6823]) that enables the convenient representation of sets of addresses such that all of the addresses in each set designate the same interface (port). For example, a 48-bit MAC (Media Access Control [RFC7042]) address, IPv4 address, and IPv6 address can be reported as all three designating the same interface. In addition, a Data Label (VLAN or Fine Grained Label (FGL [RFC7172])) is specified for the interface along with the TRILL switch and, optional the TRILL switch port, from which the interface is reachable. Such information could be used in some cases to synthesize responses to or by-pass the need for the Address Resolution Protocol (ARP [RFC826]), the IPv6 Neighbor Discovery (ND [RFC4861]) protocol, the Reverse Address Resolution Protocol (RARP [RFC903]), or the flooding of unknown destination MAC addresses [RFC7042]. If the information report is complete, it can also be used to detect and discard packets with forged source addresses.

This APPsub-TLV appears inside the TRILL GENINFO TLV specified in ESADI [RFCesadi] but may also occur in other application contexts. Directory Assisted TRILL Edge services [DirectoryScheme] are expected to make use of this APPsub-TLV.

Although, in some IETF protocols, address field types are represented by Ethertype [RFC7042] or Hardware Type [RFC5494], only Address Family Number (AFN) is used in this APPsub-TLV to represent address field type.

1.1 Conventions Used in This Document

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 [RFC2119]. Capitalized IANA Considertions terms such as "Expert Review" are to be interpreted as described in [RFC5226].

The terminology and acronyms of [RFC6325] are used herein along with the following additional acronyms and terms:

AFN: Address Family Number

APPsub-TLV: Application sub-TLV [RFC6823]

Data Label: VLAN or FGL

FGL: Fine Grained Label [RFC7172]

IA: Interface Addresses

RBridge: An alternative name for a TRILL switch

TRILL switch: A device that implements the TRILL protocol

2. Format of the Interface Addresses APPsub-TLV

The Interface Addresses (IA) APPsub-TLV is used to advertise that a set of addresses indicate the same interface (port) within a Data Label (VLAN or FGL) and to associate that interface with the TRILL switch, and optionally the TRILL switch port, by which the interface is reachable. These addresses can be in different address families. For example, it can be used to declare that a particular interface with specified IPv4, IPv6, and 48-bit MAC addresses in some particular Data Label is reachable from a particular TRILL switch.

The Template field in a particular Interface Addresses APPsub-TLV indicates the format of each Address Set it carries. Certain well-known sets of addresses are represented by special values. Other sets of addresses are specified by a list of AFNs. The Template format that uses a list of AFNs provides an explicit pattern for the type and order of addresses in each Address Set in an IA APPsub-TLV.

A device or application making use of IA APPsub-TLV data is not required to make use of all IA data. For example, a device or application that was only interested in MAC and IPv6 addresses could ignore any IPv4 or other types of address information that was present.

The figure below shows an IA APPsub-TLV as it would appear in an IS-IS PDU using an extended flooding scope [FSLSP] TLV, for example in ESADI [RFCesadi]. Within an IS-IS PDU using traditional [ISO-10589] TLVs, the Type and Length would be one byte unsigned integers equal to or less than 255.

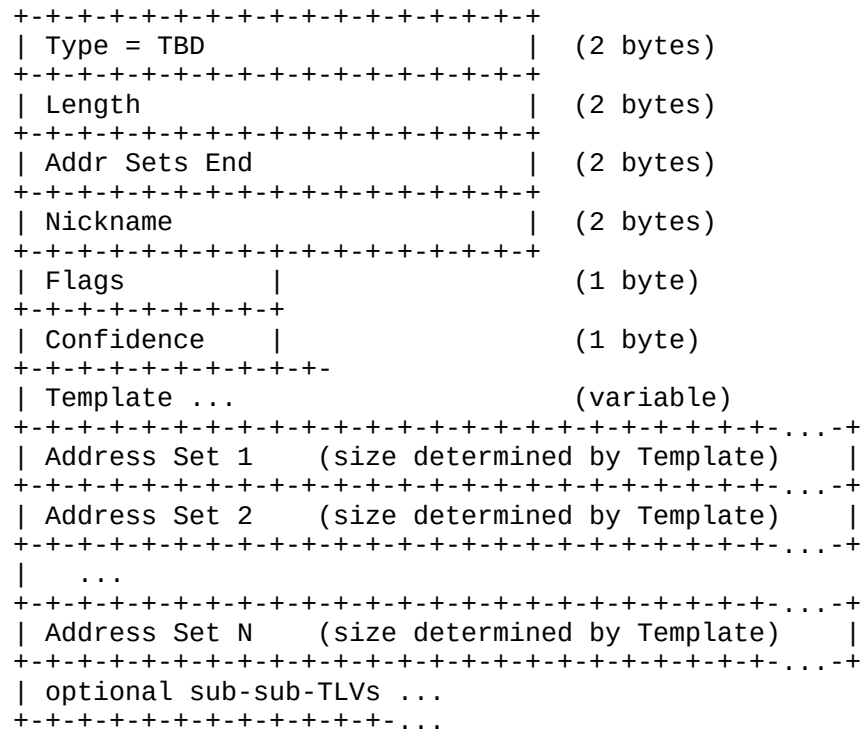


Figure 1. The Interface Addresses APPsub-TLV

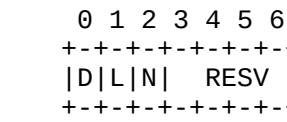
- Type: Interface Addresses TRILL APPsub-TLV type, set to TBD[#2 suggested] (IA-SUBTLV).

- Length: Variable, minimum 7. If length is 6 or less or if the APPsub-TLV extends beyond the size of an encompassing TRILL GENINFO TLV or other context, the APPsub-TLV MUST be ignored.

- Addr Sets End: The unsigned integer offset of the byte, within the IA APPsub-TLV value part, of the last byte of the last Address Set. This will be the byte just before the first sub-sub-TLV if any sub-sub-TLVs are present (see Section 3). If this is equal to Length, there are no sub-sub-TLVs. If this is greater than Length or points to before the end of the Template, the IA APPsub-TLV is corrupt and MUST be discarded. This field is always two bytes in size.

- Nickname: The nickname of the TRILL switch by which the address sets are reachable. If zero, the address sets are reachable from the TRILL switch originating the message containing the APPsub-TLV (for example, an ESADI [RFCesadi] message).

- Flags: A byte of flags as follows:



D: Directory flag: If D is one, the APPsub-TLV contains Directory information [RFC7067].

L: Local flag: If L is one, the APPsub-TLV contains information learned locally by observing ingressed frames [RFC6325]. (Both D and L can one in the same IA APPsub-TLV if a TRILL switch that had learned an address locally also advertised it as a directory.)

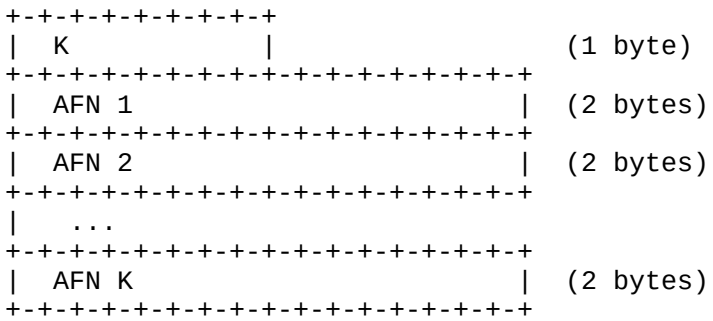
N: Notify flag: When a TRILL switch receives a new IA APPsub-TLV (one in a ESADI-LSP fragment with a higher sequence number or a new message of some other type) and the N bit is one, the TRILL switch then checks the contents of the APPsub-TLV for address sets including both an IP address and a MAC address. For each such address set it finds, a gratuitous ARP [RFC826] or spontaneous Neighbor Advertisement [RFC4861] is sent depending on whether the IP address is IPv4 or IPv6 respectively. In both cases, these are sent out all the ports of the TRILL switch that offer end station service and are in the VLAN or FGL of the address set information.

RESV: Additional reserved flag bits that MUST be sent as zero and ignored on receipt.

- Confidence: This 8-bit unsigned quantity in the range 0 to 254 indicates the confidence level in the addresses being transported [RFC6325]. A value of 255 is treated as if it was 254.

- Template: The initial byte of this field is the unsigned integer K. If K has a value from 1 to 31, it indicates that this initial byte is followed by a list of K AFNs (Address Family Numbers) that specify the exact structure and order of each Address Set occurring later in the APPsub-TLV. K can be 1, which is the minimum valid value. If K is zero, the IA APPsub-TLV is ignored. If K is 32 to 254, the length of the Template field is one byte and its value is intended to correspond to a particular ordered set of AFNs some of which are specified below. If K is 255, the length of the Template filed is three bytes and the values of the second and third byte, considered as an unsigned integer in network byte order, are reserved to correspond to future specified ordered sets of AFNs.

  If the Template uses explicit AFNs, it looks like the following.



For K in the 32 to 103 range, values indicate combinations of a specific number of MAC addresses, IPv4 addresses, IPv6 addresses, and TRILL switch port IDs appearing in that order. The value of K is

K = 32 + M + 3*v4 + 9*v6 + 36*P

where M is 0, 1, or 2 (0 if no MAC address is present, 1 if a 48-bit MAC is present, 2 if a MAC/24 (see Section 5.1) is present), v4 is the number of IPv4 addresses (limited to 0, 1, or 2) and v6 is the number of IPv6 addresses (limited to 0 through 3 inclusive), and P is the number of TRILL switch port IDs (limited to 0 or 1). That equation specifies values of K from 32 through 103. Values from 104 through 254 of the byte value are available for assignment by Expert Review (see Section 5). K = 255 indicates a three-byte Template field as specified above. All values (0 through 65,545) of this two-byte value are available for assignment by Expert Review.

If an unknown Template K value in the range 104 to 254 is received or a K of 255 followed by an unknown two byte value, the IA APPsub-TLV MUST be ignored.

- AFN: A two-byte Address Family Number. The number of AFNs present is given by K but there are no AFNs if K is greater than 31. The AFN sequence specifies the structure of the Address Sets occurring later in the TLV. For example, if Template Size is 2 and the two AFNs present are the AFNs for a 48-bit MAC and an IPv4 address, in that order, then each Address set present will consist of a 6-byte MAC address followed by a 4-byte IPv4 address. If any AFNs are present that are unknown to the receiving IS and the length of the corresponding address is not provided by a sub-sub-TLV as specified below, the receiving IS will be unable to parse the Address Sets and MUST ignore the IA APPsub-TLV.

- Address Set: Each address set in the APPsub-TLV consists of exactly the same sequence of addresses of the types specified by the Template earlier in the APPsub-TLV. No alignment, other than

to a byte boundary, is guaranteed. The addresses in each Address Set are contiguous with no unused bytes between them and the Address Sets are contiguous with no unused bytes between successive Address Sets. The Address Sets must fit within the TLV.

- o sub-sub-TLVs: If the Address Sets indicated by Addr Sets End do not completely fill the Length of the APPsub-TLV, the remaining bytes are parsed as sub-sub-TLVs [RFC5305]. Any such sub-sub-TLVs that are not known to the receiving TRILL switch are ignored. Should this parsing not be possible, for example there is only one remaining byte or an apparent sub-sub-TLV extends beyond the end of the TLV, the containing IA APPsub-TLV is considered corrupt and is ignored. (Several sub-sub-TLV types are specified in Section 3.)

Different IA APPsub-TLVs within the same or different LSPs or other data structures may have different Templates. The same AFN may occur more than once in a Template and the same address may occur in different address sets. For example, a 48-bit MAC address interface might have three different IPv6 addresses. This could be represented by an IA APPsub-TLV whose Template specifically provided for one EUI-48 address and three IPv6 addresses, which might be an efficient format if there were multiple interfaces with that pattern. Alternatively, a Template with one 48-bit MAC and one IPv6 address could be used in an IA APPsub-TLV with three address sets each having the same MAC address but different IPv6 addresses, which might be the most efficient format if only one interface had multiple IPv6 addresses and other interfaces had only one IPv6 address.

In order to be able to parse the Address Sets, a receiving TRILL switch must know at least the size of the address for each AFN or address type the Template specifies; however, the presence of the Addr Set End field means that the sub-sub-TLVs, if any, can always be located by a receiver. A TRILL switch can be assumed to know the size of the AFNs mentioned in Section 5. Should a TRILL switch wish to include an AFN that some receiving TRILL switch in the campus may not know, it SHOULD include an AFN-Size sub-sub-TLV as described in Section 3.1. If an IA APPsub-TLV is received with one or more AFNs in its template for which the receiving TRILL switch does not know the length and for which an AFN-Size sub-sub-TLV is not present, that IA APPsub-TLV MUST be ignored.

3. IA APPsub-TLV sub-sub-TLVs

IA APPsub-TLVs can have trailing sub-sub-TLVs [RFC5305] as specified below. These sub-sub-TLVs occur after the Address Sets and the amount of space available for sub-sub-TLVs is determined from the overall IA APPsub-TLV length and the value of the Addr Set End byte.

There is no ordering restriction on sub-sub-TLVs. Unless otherwise specified each sub-sub-TLV type can occur zero, one, or many times in an IA APPsub-TLV. Any sub-sub-TLVs for which the Type is unknown are ignored.

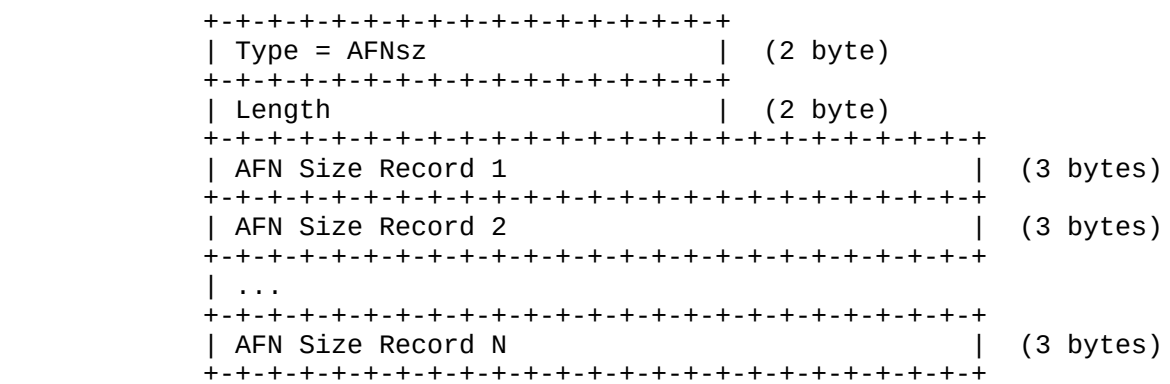
The sub-sub-TLVs data structures shown below, with two byte Types and Lengths, assume that the enclosing IA-APPsubTLV is in an extended LSP TLV [FSLSP] or some non-LSP context. If they were used in a IA-APPsubTLV in a traditional LSP [ISO-10589], the only one byte Types and Lengths could be used. As a result, any sub-sub-TLV types greater than 255 could not be used and Length would be limited to 255.

3.1 AFN Size sub-sub-TLV

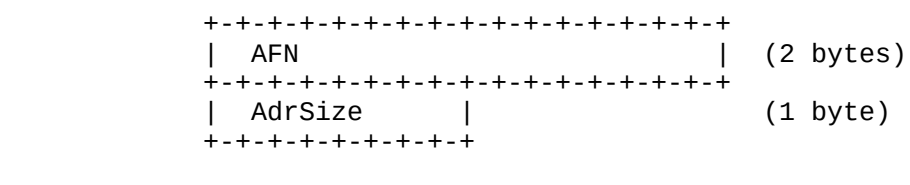
Using this sub-sub-TLV, the originating TRILL switch can specify the size of an address type. This is useful under two circumstances as follows:

1. One or more AFNs that are unknown to the receiving TRILL switch appears in the template. If an AFN Size sub-sub-TLV is present for each such AFN, then at least the IA APPsub-TLV can be parsed and possibly other addresses in each address set can still be used.

2. If an AFN occurs in the Template that represents a variable length address, this sub-sub-TLV gives its size for all occurrences in that IA APPsub-TLV.



Where each AFN Size Record is structured as follows:

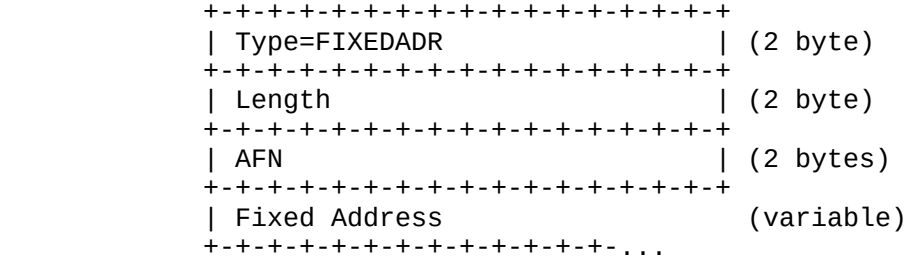


- Type: AFN-Size sub-sub-TLV type, set to 1 (AFNsz).
- Length: 3*n where n is the number of AFN Size Records present. If Length is not a multiple of 3, the sub-sub-TLV MUST be ignored.
- AFN Size Record(s): Zero or more 3-byte records, each giving the size of an address type identified by an AFN,
- AFN: The AFN whose length is being specified by the AFN Size Record.
- AdrSize: The length in bytes of addresses specified by the AFN field as an unsigned integer.

An AFN Size sub-sub-TLV for any AFN known to the receiving TRILL switch is compared with the size known to the TRILL switch. If they differ the IA APPsub-TLV is assumed to be corrupt and MUST be ignored.

3.2 Fixed Address sub-sub-TLV

There may be cases where, in an Interface Addresses APP-subTLV, the same address would appear in every address set across the APP-subTLV. To avoid wasted space, this sub-sub-TLV can be used to indicate such a fixed address. The address or addresses incorporated into the sets by this sub-sub-TLV are NOT mentioned in the IA APPsub-TLV Template.



- Type: Data Label sub-sub-TLV type, set to 2 (FIXEDADR).
- Length: variable, minimum 3. If Length is 2 or less, the sub-sub-

TLV MUST be ignored.

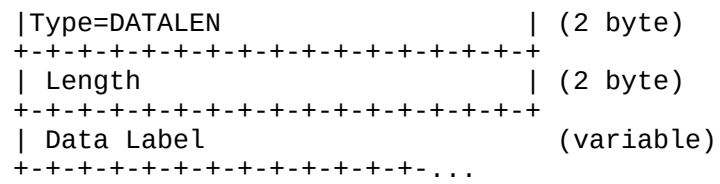
- AFN: Address Family Number of the Fixed Address.

- Fixed Address: The address of the type indicated by the preceding AFN field that is considered to be part of every Address Set in the IA APPsub-TLV.

The Length field implies a size for the Fixed Address. If that size differs from the size of the address type for the given AFN as known by the receiving TRILL switch, the Fixed Address sub-sub-TLV is considered corrupt and MUST be ignored.

3.3 Data Label sub-sub-TLV

This sub-sub-TLV indicates the Data Label within which the interfaces listed in the IA APPsub-TLV are reachable. It is useful if the IA APPsub-TLV occurs outside of the context of an ESADI [RFCesadi] or other type of message specifying the Data Label or if it is desired and permitted to override that specification. Multiple occurrences of this sub-sub-TLV indicate that the interfaces are reachable in all of the Data Labels given.



- Type: Data Label sub-TLV type, set to 3 (LABEL).

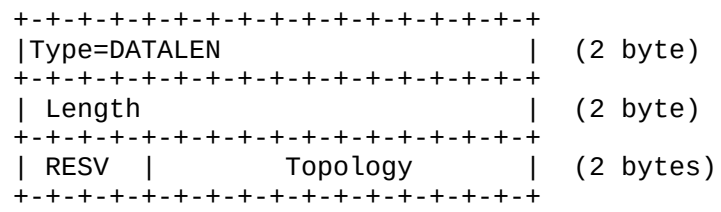
- Length: 2 or 3. If Length is some other value, the sub-sub-TLV MUST be ignored.

- Data Label: If length is 2, the bottom 12 bits of the Data Label are a VLAN ID and the top 4 bits are reserved (MUST be sent as zero and ignored on receipt). If the length is 3, the three Data Label bytes contain an FGL [RFC7172].

3.4 Topology sub-sub-TLV

The presence of this sub-sub-TLV indicates that the interfaces given in the IA APPsub-TLV are reachable in the topology give. It is useful if the IA APPsub-TLV occurs outside of the context of an ESADI

[RFCesadi] or other type of message indicating the topology or if it is desired and permitted to override that specification. If it occurs multiple times, then the Address Sets are in all of the topologies given.



- Type: Topology sub-TLV type, set to 4 (TOPOLOGY).
- Length: 2. If Length is some other values, the sub-sub-TLV MUST be ignored.

RESV: Four reserved bits. MUST be sent as zero and ignored on receipt.

- Topology: The 12-bit topology number [RFC5120].

4. Security Considerations

The integrity of address mapping and reachability information and the correctness of Data Labels (VLANs or FGLs [RFC7172]) are very important. Forged, altered, or incorrect address mapping or Data Labeling can lead to delivery of packets to the incorrect party, violating security policy. However, this document merely describes a data format and does not provide any explicit mechanisms for securing that information, other than a few trivial consistency checks that might detect some corrupted data. Security on the wire, or in storage, for this data is to be providing by the transport or storage used. For example, when transported with ESADI [RFCesadi] or RBridge Channel [RFC7178], ESADI security or Channel Tunnel [ChannelTunnel] security mechanisms can be used, respectively.

The address mapping and reachability information, if known to be complete and correct, can be used to detect some cases of forged packet source addresses [RFC7067]. In particular, if native traffic from an end station is received by a TRILL switch that would otherwise accept it but authoritative data indicates the source address should not be reachable from the receiving TRILL switch, that traffic should be discarded. The data format specified in this document may optionally include TRILL switch Port ID number so that this forged address filtering can be optionally applied with port granularity.

See [RFC6325] for general TRILL Security Considerations.

5. IANA Considerations

As specified below, IANA has allocated AFN numbers and IANA is requested to create the TRILL IS-APPsub-TLV sub-sub-TLV subregistries under the TRILL Parameters Registry.

5.1 Additional AFN Number Allocation

IANA has assigned AFN numbers as follows:

Hex	Decimal	Description	References
4007	16391	OUI	This document.
4008	16392	MAC/24	This document.
4009	16393	MAC/40	This document.
400A	16394	IPv6/64	This document.
400B	16395	RBridge Port ID	This document.

The OUI AFN is provided so that MAC addresses can be abbreviated if they have the same upper 24 bits. A MAC/24 is a 24-bit suffix intended to be pre-fixed by an OUI to create a 48-bit MAC address [RFC7042]; in the absence of an OUI, a MAC/24 entry cannot be used. A MAC/40 is a suffix intended to be pre-fixed by an OUI to create a 64-bit MAC address [RFC7042]; in the absence of an OUI, a MAC/40 entry cannot be used.

Typically, an OUI would be provided as a Fixed Address sub-sub-TLV (see Section 3.2).

After Fixed Address sub-sub-TLV processing above, each address set is processed by combining each OUI in the address set with each MAC/24 and each MAC/40 address in the address set. Depending on how many of each of these address types is present, zero or more 48-bit and/or 64-bit MAC addresses may be produced that are considered to be part of the address set. If there are no MAC/48 or MAC/40 addresses present, any OUI’s are ignored. If there are no OUIs, any MAC/24 and/or MAC/40s are ignored.

IPv6/64 is an 8-byte quantity that is the first 64 bits of an IPv6 address. IPv6/64s are ignored unless, after the processing above in this sub-section, there are one or more 48-bit and/or 64-bit MAC addresses in the address set to provide the lower 64 bits of the IPv6 address. For this purpose, an 48-bit MAC address is expanded to 64 bits as described in [RFC7042].

The following already allocated AFN values may be particularly useful for IA APPsub-TLVs:

Hex	Decimal	Description	References
0001	1	IPv4	
0002	2	IPv6	
4005	16,389	48-bit MAC	[RFC7042]
4006	16,390	64-bit MAC	[RFC7042]

Other AFNs can be found at http://www.iana.org/assignments/address-family-numbers

5.2 IA APPsub-TLV Sub-Sub-TLVs SubRegistry

IANA is requested to establish a new subregistry of the TRILL Parameter Registry for sub-sub-TLVs of the Interface Addresses APPsub-TLV with initial contents as shown below.

Name: Interface Addresses APPsub-TLV Sub-Sub-TLVs

Procedure: Expert Review

Note: Types greater than 255 are not usable in some contexts.

Reference: This document

Type	Description	Reference
0	Reserved	
1	AFN Size	This document
2	Fixed Address	This document
3	Data Label	This document
4	Topology	This document
5-254	Available	
255	Reserved	
256-65534	Available	
65535	Reserved	

Acknowledgments

The authors gratefully acknowledge the contributions and review by the following:

Linda Dunbar

The document was prepared in raw nroff. All macros used were defined within the source file.

Appendix A: Examples

Below are example IA APPsub-TLVs.

A.1 Simple Example

Below is an annotated IA APPsub-TLV carrying two simple pairs of EUI-48 MAC addresses and IPv4 addresses from a Push Directory [RFC7042]. No sub-sub-TLVs are included.

```
0x0002(TBD)   Type: Interface Addresses
0x001B        Length: 27 (=0x1B)
0x001B        Address Sets End: 27 (=0x1B)
0x1234        RBridge Nickname from which reachable
0b10000000    Flags: Push Directory data
0xE3          Confidence = 227
35            Template: 35 (0x23) = 32 + 1(MAC48) + 3*1(IPv4)

      Address Set One
0x00005E0053A9   48-bitMAC address
198.51.100.23    IPv4 address

      Address Set Two
0x00005E00536B   48-bit MAC address
203.0.113.201    IPv4 address
```

Size includes 7 for the fixed fields though and including the one byte template, plus 2 times the Address Set size. Each Address Set is 10 bytes, 6 for the 48-bit MAC address plus 4 for the IPv4 address. So total size is 7 + 2*10 = 27.

See Section 2 for more information on Template.

A.2 Complex Example

Below is an annotated IA APPsub-TLV carrying three sets of addresses, each consisting of an EUI-48 MAC address, an IPv4 addresses, an IPv6 address, and an RBridge Port ID, all from a Push Directory [RFC7042]. The IPv6 address for each address set is synthesized from the MAC address given in that set and the IPv6/64 64-bit prefix provided through a Fixed Address sub-sub-TLV. In addition, a sub-sub-TLV is included that provides an FGL which overrides whatever Data Label may be provided by the envelope (for example ESADI [RFCesadi]) within which this IA APPsub-TLV occurs.

```
0x0002(TBD)   Type: Interface Addresses
0x0036        Length: 54 (=0x36)
0x0021        Address Sets End: 33 (=0x21)
0x4321        RBridge Nickname from which reachable
0b10000000    Flags: Push Directory data
0xD3          Confidence = 211
72            Template: 72(0x48)=32+1(MAC48)+3*1(IPv4)+36*1(P)

      Address Set One
0x00005E0053DE   48-bitMAC address
198.51.100.105   IPv4 address
0x1DE3           RBridge Port ID

      Address Set Two
0x00005E0053E3   48-bit MAC address
203.0.113.89     IPv4 address
0x1DEE           RBridge Port ID

      Address Set Three
0x00005E0053D3   48-bit MAC address
192.0.2.139      IPv4 address
0x01DE           RBridge Port ID

      sub-sub-TLV One
0x0003           Type: Data Label
0x0003           Length: implies FGL
0xD3E3E3         Fine Grained Label

      sub-sub-TLV Two
0x0002           Type: Fixed Address
0x000A           Size: 0x0A = 10
0x400A           AFN: IPv6/64
0x20010DB800000000   IPv6 Prefix: 2001:DB8::
```

See Section 2 for more information on Template.

The Fixed Address sub-sub-TLV causes the IPv6/64 value give to be treated as if it occurred as a 4th entry inside each of the three Address Sets. When there is an IPv6/64 entry and a 48-bit MAC entry, the MAC value is expanded by inserting 0xFFFE immediately after the OUI and the resulting 64-bit value is used as the lower 64 bits of the resulting IPv6 address [RFC7042]. As a result, a receiving TRILL switch would treat the three Address Sets shown as if they had an IPv6 address in them as follows:

```
      Address Set One
0x20010DB80000000000000005EFFFE0053DE  IPv6 Address

      Address Set Two
0x20010DB80000000000000005EFFFE0053E3  IPv6 Address

      Address Set Three
0x20010DB80000000000000005EFFFE0053D3  IPv6 Address
```

As an alternative to the compact "well know value" Template encoding used in this example above, the less compact explicit AFN encoding could have been used. In that case, the IA APPsub-TLV would have started as follows:

```
0x0002(TBD)   Type: Interface Addresses
0x003C        Length: 60 (=0x3C)
0x0027        Address Sets End: 39 (=0x27)
0x4321        RBridge Nickname from which reachable
0b10000000    Flags: Push Directory data
0xD3          Confidence = 211
0x3           Template: 3 AFNs
0x4005        AFN: 48-bit MAC
0x0001        AFN: IPv4
0x400B        AFN: RBridge Port ID
```

As a final point, since the 48-bit MAC addresses in these three Address Sets all have the same OUI (the IANA OUI [RFC7042]), it would have been possible to just have a MAC/24 value giving the lower 24 bits of the MAC in each Address Set. The OUI would them be supplied by a second Fixed Address sub-sub-TLV proving the OUI. With N Address Sets, this would have saved 3*N or 9 bytes in this case at the cost of 7 bytes (1 each for the type and length of the sub-sub-TLV, 2 for the OUI AFN number, and 3 for the OUI). So, even with just three Address Sets, there would be a small net saving of 2 bytes. The savings would grow with a larger number of Address Sets.

Appendix Z: Change History

From -00 to -01

1. Update references for RFC publications.
2. Add this Change History Appendix.

Normative References

[ISO-10589] - ISO/IEC 10589:2002, Second Edition, "Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)", 2002.

[RFC826] - Plummer, D., "An Ethernet Address Resolution Protocol", RFC 826, November 1982.

[RFC903] - Finlayson, R., Mann, T., Mogul, J., and M. Theimer, "A Reverse Address Resolution Protocol", STD 38, RFC 903, June 1984.

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

[RFC4861] - Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5120] - Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.

[RFC5226] - Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5305] - Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC6325] - Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

[RFC6823] - Ginsberg, L., Previdi, S., and M. Shand, "Advertising Generic Information in IS-IS", RFC 6823, December 2012.

[RFC7042] - Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 7042, October 2013.

[RFC7172] - Eastlake 3rd, D., Zhang, M., Agarwal, P., Perlman, R., and D. Dutt, "Transparent Interconnection of Lots of Links (TRILL): Fine-Grained Labeling", RFC 7172, May 2014.

[FSLSP] - Ginsberg, L., S. Previdi, Y. Yang, "IS-IS Flooding Scope LSPs", draft-ietf-isis-fs-lsp, work in progress.

Informational References

[ARP reduction] - Shah, et. al., "ARP Broadcast Reduction for Large Data Centers", Oct 2010.

[ChannelTunnel] - D. Eastlake, Y. Li, "TRILL: RBridge Channel Tunnel Protocol", draft-eastlake-trill-channel-tunnel, work in progress.

[DirectoryScheme] - Dunbar, L., D. Eastlake, R. Perlman, I. Gashinsky, Y. Li, "TRILL": Directory Assistance Mechanisms", draft-dunbar-trill-scheme-for-directory-assist, work in progress.

[RFC5494] - Arkko, J. and C. Pignataro, "IANA Allocation Guidelines for the Address Resolution Protocol (ARP)", RFC 5494, April 2009.

[RFC7067] - Dunbar, L., Eastlake 3rd, D., Perlman, R., and I. Gashinsky, "Directory Assistance Problem and High-Level Design Proposal", RFC 7067, November 2013.

[RFC7178] - Eastlake 3rd, D., Manral, V., Li, Y., Aldrin, S., and D. Ward, "Transparent Interconnection of Lots of Links (TRILL): RBridge Channel Support", RFC 7178, May 2014.

[RFCesadi] - Zhai, H., F. Hu, R. Perlman, D. Eastlake, O. Stokes, "TRILL (Transparent Interconnection of Lots of Links): The ESADI (End Station Address Distribution Information) Protocol", draft-ietf-trill-esadi, work in progress.

Authors' Addresses

Donald Eastlake
Huawei Technologies
155 Beaver Street
Milford, MA 01757 USA

Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012 China

Phone: +86-25-56622310
Email: liyizhou@huawei.com

Radia Perlman
Intel Labs
2200 Mission College Blvd.
Santa Clara, CA 95054-1549 USA

Phone: +1-408-765-8080
Email: Radia@alum.mit.edu

Copyright, Disclaimer, and Additional IPR Provisions

TRILL Working Group
Internet Draft
Intended Status: Standard Track

Deepak Kumar
Samer Salam
Tissa Senevirathne
Cisco

Expires Jan 2015

July 26, 2014

TRILL OAM MIB
draft-ietf-trill-oam-mib-01.txt

Status of this Memo

Copyright Notice

Abstract

This document specifies the Management Information Base (MIB) for the IETF TRILL (Transparent Interconnection of Lots of Links) OAM objects.

Table of Contents

1. Introduction

Overall, TRILL OAM is intended to meet the requirements given in [RFC6905]. The general framework for TRILL OAM is specified in [TRILLOAMFRM]. The details of the Fault Management [FM] solution, conforming to that framework, are presented in [TRILLOAMFM]. The solution leverages the message format defined in Ethernet Connectivity Fault Management (CFM) [802.1Q] as the basis for the TRILL OAM message channel.

This document uses the CFM MIB modules defined in [802.1Q] as the basis for TRILL OAM MIB, and augments the existing tables to add new TRILL managed objects required by TRILL. This document further specifies a new table with associated managed objects for TRILL OAM specific capabilities.

2. The Internet-Standard Management Framework

For a detailed overview of the Internet-Standard Management Framework, please refer to [RFC3410]. Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the Structure of Management Information (SMI) specification. This memo specifies a MIB module that is compliant to SMIv2 [RFC2578], [RFC2579] and [RFC2580].

3. Overview

The TRILL-OAM-MIB module is intended to provide an overall framework for managing TRILL OAM. It leverages the IEEE8021-CFM-MIB and IEEE8021-CFM-V2-MIB modules defined in [802.1Q], and augments the Mep and Mep Db entries. It also adds a new table for TRILL OAM specific messages.

4. Conventions

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].

5. Structure of the MIB module

Objects in this MIB module are arranged into subtrees. Each subtree is organized as a set of related objects. The various subtrees are shown below, supplemented with the required elements of the IEEE8021-CFM-MIB module.

5.1. Textual Conventions

Textual conventions are defined to represent object types relevant to the TRILL OAM MIB.

5.2. TRILL-OAM-MIB relationship to IEEE8021-TC-MIB

In TRILL, traffic labeling can be done using either a 12-bit VLAN or a 24-bit fine grain label [RFCfg1].

IEEE8021-TC-MIB defines IEEE8021ServiceSelectorType with two values:

- 1 representing a vlanId, and

- 2 representing a 24 bit isid.

We propose to use value 2 for TRILL’s fine grain label. As such, TRILL-OAM-MIB will import IEEE8021ServiceSelectorType,

IEEE8021ServiceSelectorValueOrNone, and IEEE8021ServiceSelectorValue from IEEE8021-TC-MIB.

5.3. TRILL OAM MIB Tree

```
TRILL-OAM-MIB

  |--trillOamNotifications

     |--trillOamFaultAlarm

  |--trillOamMibObjects

     |--trillOamMep

        |--trillOamMepTable

        |--trillOamMepFlowCfgTable

        |--trillOamPtrTable

        |--trillOamMtrTable

        |--trillOamMepDbTable
```

5.3.1. Notifications

Notification (fault alarm) is sent to the management entity with the OID of the MEP that has detected the fault.

5.3.2. TRILL OAM MIB Per MEP Objects

The TRILL OAM MIB Per MEP Objects are defined in the trillOamMepTable. The trillOamMepTable augments the dot1agCfmMepEntry (please see section 6.1) defined in IEEE8021-CFM-MIB. It includes objects that are locally defined for an individual MEP and its associated Flow.

5.3.2.1. trillOamMepTable Objects

- o trillOamMepRName - This object contains the Rbridge Nickname as defined in [RFC6325] section 3.7.

- o trillOamMepPtmTId - indicates the next sequence number/transaction identifier to be sent in a Path Trace message. The sequence number may be zero because it wraps around.

o trillOamMepNexttMtmTId - indicates the next sequence number/transaction identifier to be sent in a Multi-destination message. The sequence number may be zero because it wraps around.

o trillOamMepMepPtrIn - indicates the total number of valid, in-order, Path Trace Replies received.

o trillOamMepPtrInOutofOrder - indicates the total number of valid, out-of-order, Path Trace Replies received.

o trillOamMepPtrOut - indicates the total number of valid Path Trace Replies transmitted.

o trillOamMepMtrIn - indicates the total number of valid, in-order, Multi-destination Replies received.

o trillOamMepMtrInOutofOrder - indicates the total number of valid, out-of-order, Multi-destination Replies received.

o trillOamMepMtrOut - indicates the total number of valid Multi-destination Replies transmitted.

o trillOamMepTxLbmDestRName - indicates the target destination Rbridge NickName as defined in [RFC6325] section 3.7.

o trillOamMepTxLbmHC - indicates the hop count field to be transmitted.

o trillOamMepTxLbmReplyModeOob - True indicates that the Reply Mode of the Loopback message is requested to be out-of-band, and that the "Out of band IP address" TLV is to be transmitted. False indicates that in-band reply is transmitted.

o trillOamMepTransmitLbmReplyIp - indicates the IP address to be transmitted in the "Out of band IP Address TLV" in the Loopback message.

o trillOamMepTxLbmFlowEntropy - indicates the 128 bytes Flow entropy to be transmitted, as defined in [TRILLOAMFM].

o trillOamMepTxPtmDestRName - indicates the target Destination Rbridge Nickname to be transmitted, as defined in [RFC6325] section 3.7.

o trillOamMepTxPtmHC - indicates the hop count field to be transmitted.

o trillOamMepTxPtmReplyModeOob - True indicates that the Reply Mode of the Path Trace message is requested to be out-of-band, and that the "Out of band IP address TLV" is to be transmitted. False indicates that in-band reply is transmitted.

o trillOamMepTransmitPtmReplyIP - indicates the IP address to be transmitted in the "Out of band IP Address TLV" in the Path Trace message.

o trillOamMepTranmitPtmFlowEntropy - indicates the 128 bytes Flow entropy to be transmitted, as defined in [TRILLOAMFM].

o trillOamMepTxPtmStatus - A Boolean flag set to True by the MEP Path Trace Initiator State Machine or a MIB manager to indicate that another Path trace message is being transmitted. Reset to false by the MEP Initiator State Machine.

o trillOamMepTxPtmResultOK - Indicates the result of the operation, True : The Path Trace Message(s) will be (or has been) sent, False: The Path Trace Message(s) will not be sent.

o trillOamMepTxPtmMessages - The number of Path Trace messages to be transmitted.

o trillOamMepTxPtmSeqNumber - Indicates the Path Trace Transaction Identifier of the first PTM (to be) sent. The value returned is undefined if trillOamMepTxPtmResultOK is false.

o trillOamMepTxMtmTree - Indicates the Multi-destination Tree identifier as defined in RFC6325.

o trillOamMepTxMtmHC - Indicates the hop count field to be transmitted.

o trillOamMepTxMtmReplyModeOob - True indicates that the Reply of the Multi-destination message is requested to be out-of-band, and that the "Out of band IP address TLV" is to be transmitted. False indicates that in-band reply is transmitted.

o trillOamMepTransmitMtmReplyIp - the IP address to be transmitted in the "Out of band IP address TLV" in the Multi-destination message.

o trillOamMepTxMtmFlowEntropy - 128 Byte Flow Entropy to be transmitted, as defined in [TRILL-FM].

o trillOamMepTxMtmStatus - A Boolean flag set to True by the MEP Multi-Destination Initiator State Machine or a MIB manager

to indicate that another Multicast trace message is being transmitted. Reset to False by the MEP Initiator State Machine.

o trillOamMepTxMtmResultOK - Indicates the result of the operation: -True The Multi-destination Message(s) will be (or has been) sent. -False The Multi-destination Message(s) will not be sent.

o trillOamMepTxMtmMessages -The number of Multi-Destination Messages to be transmitted.

o trillOamMepTxMtmSeqNumber - The Sequence Number of the first Multi-destination message (to be) sent. The value returned is undefined if trillOamMepTxMtmResultOK is false.

o trillOamMepTxMtmScopeList - The Multi-destination Rbridge Scope list, 2 octets per Rbridge.

5.3.2.2. trillOamMepFlowCfgTable Objects

Each row in this table represents a Flow Configuration Entry for the associated MEP. The table uses four indices. The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific Flow Configuration Entry on the selected MEP. Some write-able objects in this table are only applicable in certain cases (as described under each object below), and attempts to write values for them in other cases will be ignored.

o trillOamMepFlowCfgIndex - an index to the TRILL OAM Mep flow configuration table which indicates the specific Flow for the MEP. The index is never reused for other flow sessions on the same MEP while this session is active. The index value keeps increasing until it wraps to 0. This value can also be used in Flow-identifier TLV.

o trillOamMepFlowCfgFlowEntropy - This is 96 bytes of flow entropy as described in [TRILL-FM].

o trillOamMepFlowCfgDestRname - The target Rbridge nickname field to be transmitted as defined in [RFC6325] section 3.7.

o trillOamMepFlowCfgFlowHC - indicates the time to live field to be transmitted.

o trillOamMepFlowCfgRowStatus - indicates the status of row. The write-able columns in a row cannot be changed if the row is active. All columns MUST have a valid value before a row can be

activated.

5.3.2.3. trillOamPtrTable Objects

Each row in the table represents a Path Trace Reply Entry for the defined MEP and Transaction. This table uses four indices. The first three indices identify the MEP and the fourth index specifies the Transaction Identifier, and this transaction identifier uniquely identifies the response for a MEP which can have multiple flow.

o trillOamMepPtrTransactionId - indicates Transaction identifier/sequence number returned by a previous transmit path trace message command, indicating which PTM’s response is going to be returned.

o trillOamPtrHC - indicates hop count field value for a returned PTR.

o trillOamMepPtrFlag - indicates FCOI field value for a returned PTR.

o trillOamMepPtrErrorcode - indicates the Return code and Return sub-code value for a returned PTR.

o trillOamMepPtrTerminalMep - indicates a Boolean value stating whether the forwarded PTM reached a MEP enclosing its MA, as returned in the Terminal MEP flag field.

o trillOamMepPtrNextEgressIdentifer -An integer field holding the last Egress Identifier returned in the PTR Upstream Rbridge nickname TLV of the PTR. The Last Egress identifies the Upstream Nickname.

o trillOamMepPtrIngress - The value returned in the Ingress Action field of the PTM. The value ingNoTlv(0) indicates that no Reply Ingress TLV was returned in the PTM.

o trillOamMepPtrIngressMac - indicates the MAC address returned in the ingress MAC address filed.

o trillOamMepIngressPortIdSubtype - indicates ingress Port ID. The format of this object is determined by the value of the trillOamMepPtrIngressPortIdSubtype object.

o trillOamMepIngressPortId - indicates the ingress port ID. The format of this object is determined by the value of the trillOamMepPtrIngressPortId object.

o trillOamMepPtrEgressPortIdSubtype - indicates the value returned in the Egress Action field of the PTM. The value ingNoTlv(0) indicates that no Reply Egress TLV was returned in the PTM.

o trillOamMepPtrEgressPortId - indicates the egress port ID. The format of this object is determined by the value of trillOamMepPtrEgressPortId object.

o trillOamMepPtrChassisIdSubtype - This object specifies the format for the Chassis ID returned in the Sender ID TLV of the PTR, if any. This value is ignored if the trillOamMepPtrChassiId has a length of 0.

o trillOamMepPtrChassisId - indicates the chassis ID returned in the Sender ID TLV of the PTR, if any. The format of this object is determined by the value of the trillOamMepPtrChassisIdSubtype object.

o trillOamMepPtrOrganizationSpecificTlv - indicates all Organization specific TLVs returned in the PTR, if any. Includes all octets including and following the TLV length field of each TLV, concatenated together.

o trillOaMepPtrNextHopNicknames - indicates Next hop Rbridge List TLV returned in the PTR, if any. Includes all octets including and following the TLV length concatenated together.

5.3.2.4. trillOamMtrTable Objects

This table includes Multi-destination Reply managed objects. Each row in the table represents a Multi-destination Reply Entry for the defined MEP and Transaction. This table uses five indices: The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific Transaction Identifier on the selected MEP. The fifth index is the receive order of Multi-destination replies. Some write-able objects in this table are only applicable in certain cases (as described under each object below), and attempts to write a value for them in other cases will be ignored.

o trillOamMepMtrTransactionId - indicates Transaction identifier/sequence number returned by a previous transmit Multi-destination message command, indicating which MTM’s response is going to be returned.

o trillOamMepMtrReceiveOrder - indicates an index to

distinguish among multiple MTR with same same MTR Transaction Identifier field value. trillOamMepMtrReceiveOrder are assigned sequentially from 1, in the order that the Multi-destination Tree Initiator received the MTRs.

o trillOamMepMtrFlag - indicates FCOI field value for a returned MTR.

o trillOamMepMtrErrorCode - indicates return code and return sub code value for a returned MTR.

o trillOamMepMtrLastEgressIdentifier - indicates an integer field holding the Last Egress Identifier returned in the MTR Upstream Rbridge Nickname TLV of the MTR. The Last Egress Identifier identifies the Upstream Nickname.

o trillOamMepMtrIngress - indicates the value returned in the Ingress Action Field of the MTR. The value ingNoTlv(0) indicates that no Reply Ingress TLV was returned in the MTM.

o trillOamMepMtrIngressMac - indicates the MAC address returned in the ingress MAC address field.

o trillOamMepMtrIngressPortIdSubtype - indicates the ingress Port ID. The format of this object is determined by the value of the trillOamMepMtrIngressPortIdSubtype object.

o trillOamMepMtrIngressPortId - indicates the ingress Port Id. The format of this object is determined by the value of the trillOamMepMtrIngressPortId object.

o trillOamMepMtrEgress - indicates the value returned in the Egress Action field of the MTR. The value ingNoTLv(0) indicates that no Reply Egress TLV was returned in the MTR.

o trillOamMepMtrEgressMac - indicates the MAC address returned in the egress MAC address field.

o trillOamMepMtrEgressPortIdSubtype - indicates the egress Port ID. The format of this object is determined by the value of the trillOamMepMtrEgressPortIdSubtype object.

o trillOamMepMtrEgressPortId - indicates the egress port ID. The format of this object is determined by the value of the trillOamMepMtrEgressPortId object.

o trillOamMepMtrChassisIdSubtype - indicates the format of the chassis ID returned in the Sender ID TLV of the MTR, if any.

The value is ignored if the trillOamMepMtrChassisId has length of 0.

o trillOamMepMtrChassisId - indicates the chassis ID returned in the Sender ID TLV of the MTR, if any. The format of this object is determined by the value of the trillOamMepMtrChassisIdSubtype object.

o trillOamMepMtrOrganizationSpecificTlv - indicates all Organization specific TLVs returned in the MTR, if any. Includes all octets including and following the TLV length filed of each TLV, contacted together.

o trillOamMepMtrNextHopNicknames - indicates next hop Rbridge List TLV returned in the PTR, if any. Includes all octets including and following the TLV length filed of each TLV, concatenated together.

o trillOamMepMtrNextHopTotalReceivers - indicates value indicating that MTR response contains Multicast receiver availability TLV.

o trillOamMepMtrReceiverCount - indicates the number of Multicast receivers available on responding Rbridge on the VLAN specified by the diagnostic VLAN.

5.3.2.4. trillOamMepDbTable Objects

This table is an augmentation of the dot1agCfmMepDbTable, and rows are automatically added or deleted from this table based upon row creation and destruction of the dot1agCfmMepDbTable.

o trillOamMepDbFlowIndex - This object identifies the Flow. If the Flow Identifier TLV is received than index received can also be used.

o trillOamMepCfgFlowEntropy - indicates 96 bytes of Flow entropy.

o trillOamMepDbFlowState - indicates the operational state of the remote MEP (flow based) IFF state machines.

o trillOamMepDbRmepFailedOkTime - indicates the time (sysUpTime) at which the Remote Mep Flow State machine last entered either the RMEP_FAILED or RMEP_OK state.

o trillOamMepDbRbridgeName - indicates Remote MEP Rbridge Nickname.

6. Relationship to other MIB module

The IEEE8021-CFM-MIB, IEEE801-CFM-V2-MIB and LLDP-MIB contain objects relevant to TRILL OAM MIB. Management objects contained in these modules are not duplicated here, to reduce overlap to the extent possible.

6.1. Relationship to IEEE8021-CFM-MIB

TRILL OAM MIB Imports the following management objects from IEEE8021-CFM-MIB:

- dot1agCfmMdIndex
- dot1agCfmMaIndex
- dot1agCfmMepIdentifier
- dot1agCfmMepEntry
- dot1agCfmMepDbEntry
- Dot1agCfmIngressActionFieldValue
- Dot1agCfmEgessActionFieldValue
- Dot1agCfmRemoteMepState

trillOamMepTable Augments dot1agCfmMepEntry. Implementation of IEEE-CFM-MIB is required as we are Augmenting the IEEE-CFM-MIB Table. Objects/Tables that are not applicable to a TRILL implementation have to be handled by the TRILL implementation back end and appropriate values as described in IEEE-CFM-MIB have to be returned.

6.2. MIB modules required for IMPORTS

The following MIB module IMPORTS objects from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], IEEE-8021-CFM-MIB, LLDP-MIB.

7. Definition of the TRILL OAM MIB module

```
TRILL-OAM-MIB DEFINITIONS ::= BEGIN

IMPORTS

    MODULE-IDENTITY,
```

```
        OBJECT-TYPE,
        NOTIFICATION-TYPE,
        Counter32,
        Unsigned32,
        Integer32
           FROM SNMPv2-SMI
        RowStatus,
        TruthValue,
        TimeStamp,
        MacAddress
            FROM SNMPv2-TC
        OBJECT-GROUP,
        NOTIFICATION-GROUP,
        MODULE-COMPLIANCE
            FROM SNMPv2-CONF
        dot1agCfmMdIndex,
        dot1agCfmMaIndex,
        dot1agCfmMepIdentifier,
        dot1agCfmMepEntry,
        dot1agCfmMepDbEntry,
        Dot1agCfmIngressActionFieldValue,
        Dot1agCfmEgressActionFieldValue,
        Dot1agCfmRemoteMepState
            FROM IEEE8021-CFM-MIB
        LldpChassisId,
        LldpChassisIdSubtype,
        LldpPortId
            FROM LLDP-MIB;

    trillOamMib MODULE-IDENTITY
        LAST-UPDATED    "201407261200Z"
        ORGANIZATION    "TBD"
        CONTACT-INFO
                "E-mail:  dekumar@cisco.com
                Postal:  510 McCarthy Blvd
                         Milpitas, CA 95035
                         U.S.A.
                Phone:   +1 408 853 9760"
        DESCRIPTION
            "This MIB module contains the management objects for the
            management of Trill Services Operations, Administration
            and Maintenance.
            Initial version. Published as RFC xxxx.

            ----------------------------------------------------------------
            Reference Overview

            A number of base documents have been used to create the
```

Textual Conventions MIB. The following are the abbreviations for the baseline documents:

[CFM] refers to 'Connectivity Fault Management', IEEE 802.1ag-2007, December 2007
[Q.840.1] refers to 'ITU-T Requirements and analysis for NMS-EMS management interface of Ethernet over Transport and Metro Ethernet Network (EoT/MEN)', March 2007
[Y.1731] refers to ITU-T Y.1731 'OAM functions and mechanisms for Ethernet based networks', February 2011

Abbreviations Used

Term	Definition
CCM	Continuity Check Message
CFM	Connectivity Fault Management
CoS	Class of Service
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ITU-T	International Telecommunication Union - Telecommunication Standardization Bureau
MAC	Media Access Control
MA	Maintenance Association (equivalent to a MEG)
MD	Maintenance Domain (equivalent to a OAM Domain in MEF 17)
MD Level	Maintenance Domain Level (equivalent to a MEG level)
ME	Maintenance Entity
MEG	Maintenance Entity Group (equivalent to a MA)
MEG Level	Maintenance Entity Group Level (equivalent to MD Level)
MEP	Maintenance Association End Point or MEG End Point
MIB	Management Information Base
MIP	Maintenance Domain Intermediate Point or MEG Intermediate Point
MP	Maintenance Point. One of either a MEP or a MIP
OAM	Operations, Administration, and Maintenance On-Demand OAM actions that are initiated via manual intervention for a limited time to carry out diagnostics. On-Demand OAM can result in singular or periodic OAM actions during the diagnostic time interval
PDU	Protocol Data Unit
RFC	Request for Comment
SNMP	Simple Network Management Protocol
SNMP Agent	An SNMP entity containing one or more command responder and/or notification originator applications (along with their associated SNMP engine). Typically implemented in an NE.
SNMP Manager	An SNMP entity containing one or more command generator

```
            and/or notification receiver applications (along with
            their associated SNMP engine). Typically implemented in
            an EMS or NMS.
           TLV          Type Length Value, a method of encoding Objects
           UTC          Coordinated Universal Time
           UNI          User-to-Network Interface
           VLAN         Virtual LAN"
        REVISION        "201407261200Z"
        DESCRIPTION
            "Initial version. Published as RFC xxxx."
        ::= { mib-2 xxx }


   -- RFC Ed.: assigned by IANA, see section 9 for details
   --
   -- ************************************************************
   -- Object definitions in the TRILL OAM MIB Module
   -- ************************************************************

   trillOamNotifications  OBJECT IDENTIFIER
       ::= { trillOamMib 0 }

   trillOamMibObjects  OBJECT IDENTIFIER
       ::= { trillOamMib 1 }

   trillOamMibConformance  OBJECT IDENTIFIER
       ::= { trillOamMib 2 }

   -- ************************************************************
   -- Groups in the TRILL OAM MIB Module
   -- ************************************************************

   trillOamMep  OBJECT IDENTIFIER
       ::= { trillOamMibObjects 1 }


   -- ************************************************************
   -- TRILL OAM MEP Configuration
   -- ************************************************************

   trillOamMepTable OBJECT-TYPE
       SYNTAX          SEQUENCE OF TrillOamMepEntry
       MAX-ACCESS      not-accessible
       STATUS          current
       DESCRIPTION
           "This table is an extension of the dot1agCfmMepTable and rows
           are automatically added or deleted from this table based upon
           row creation and destruction of the dot1agCfmMepTable.
```

```
            This table represents the local MEP TRILL OAM configuration
            table. The primary purpose of this table is provide local
            parameters for the TRILL OAM function found in [TRILL-FM] and
            instantiated at a MEP."
        REFERENCE       "[TRILL-FM]"
        ::= { trillOamMep 1 }

    trillOamMepEntry OBJECT-TYPE
        SYNTAX          TrillOamMepEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION
            "The conceptual row of trillOamMepTable."
        AUGMENTS           { dot1agCfmMepEntry  }
        ::= { trillOamMepTable 1 }

    TrillOamMepEntry ::= SEQUENCE {
            trillOamMepRName              Unsigned32,
            trillOamMepNextPtmTId         Unsigned32,
            trillOamMepNextMtmTId         Unsigned32,
            trillOamMepPtrIn              Counter32,
            trillOamMepPtrInOutofOrder    Counter32,
            trillOamMepPtrOut             Counter32,
            trillOamMepMtrIn              Counter32,
            trillOamMepMtrInOutofOrder    Counter32,
            trillOamMepMtrOut             Counter32,
            trillOamMepTxLbmDestRName     Unsigned32,
            trillOamMepTxLbmHC           Unsigned32,
            trillOamMepTxLbmReplyModeOob  TruthValue,
            trillOamMepTransmitLbmReplyIp OCTET STRING,
            trillOamMepTxLbmFlowEntropy   OCTET STRING,
            trillOamMepTxPtmDestRName     Unsigned32,
            trillOamMepTxPtmHC           Unsigned32,
            trillOamMepTxPtmReplyModeOob  TruthValue,
            trillOamMepTransmitPtmReplyIp OCTET STRING,
            trillOamMepTxPtmFlowEntropy   OCTET STRING,
            trillOamMepTxPtmStatus        TruthValue,
            trillOamMepTxPtmResultOK      TruthValue,
            trillOamMepTxPtmMessages      Integer32,
            trillOamMepTxPtmSeqNumber     Unsigned32,
            trillOamMepTxMtmTree          Unsigned32,
            trillOamMepTxMtmHC           Unsigned32,
            trillOamMepTxMtmReplyModeOob  TruthValue,
            trillOamMepTransmitMtmReplyIp OCTET STRING,
            trillOamMepTxMtmFlowEntropy   OCTET STRING,
            trillOamMepTxMtmStatus        TruthValue,
            trillOamMepTxMtmResultOK      TruthValue,
            trillOamMepTxMtmMessages      Integer32,
```

```
            trillOamMepTxMtmSeqNumber     Unsigned32,
            trillOamMepTxMtmScopeList     OCTET STRING
    }

    trillOamMepRName OBJECT-TYPE
        SYNTAX          Unsigned32 (0..65471)
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "This object contains Rbridge NickName of TRILL Rbridge as
            defined in RFC 6325 section 3.7."
        REFERENCE "TRILL-FM and RFC 6325 section 3.7"
        ::= { trillOamMepEntry 1 }

    trillOamMepNextPtmTId OBJECT-TYPE
        SYNTAX          Unsigned32
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Next sequence number/transaction identifier to be sent in a
            Path Trace message. This sequence number can be zero because it
            wraps around. Implementation should be unique to identify
            Transaction Id for a MEP with multiple flows."
        REFERENCE "TRILL-FM 10.1.1"
        ::= { trillOamMepEntry 2 }

    trillOamMepNextMtmTId OBJECT-TYPE
        SYNTAX          Unsigned32
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Next sequence number/transaction identifier to be sent in a
            Multi-destination message. This sequence number can be zero
            because it wraps around. Implementation should be unique to
            identify Transaction Id for a MEP with multiple flows."
        REFERENCE "TRILL-FM 11.2.1"
        ::= { trillOamMepEntry 3 }

    trillOamMepPtrIn OBJECT-TYPE
        SYNTAX          Counter32
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Total number of valid, in-order Path Trace Replies received."
        REFERENCE "TRILL-FM section 10"
        ::= { trillOamMepEntry 4 }

    trillOamMepPtrInOutofOrder OBJECT-TYPE
```

```
        SYNTAX          Counter32
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Total number of valid, out-of-order Path Trace Replies received."
        REFERENCE "TRILL-FM section 10"
        ::= { trillOamMepEntry 5 }

    trillOamMepPtrOut OBJECT-TYPE
        SYNTAX          Counter32
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Total number of valid, Path Trace Replies transmitted."
        REFERENCE "TRILL-FM section 10"
        ::= { trillOamMepEntry 6 }

    trillOamMepMtrIn OBJECT-TYPE
        SYNTAX          Counter32
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Total number of valid, in-order Multi-destination Replies
            received."
        REFERENCE "TRILL-FM section 11"
        ::= { trillOamMepEntry 7 }

    trillOamMepMtrInOutofOrder OBJECT-TYPE
        SYNTAX          Counter32
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Total number of valid, out-of-order Multi-destination Replies
            received."
        REFERENCE "TRILL-FM section 11"
        ::= { trillOamMepEntry 8 }

    trillOamMepMtrOut OBJECT-TYPE
        SYNTAX          Counter32
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Total number of valid, Multi-destination Replies
            transmitted."
        REFERENCE "TRILL-FM section 11"
        ::= { trillOamMepEntry 9 }

    trillOamMepTxLbmDestRName OBJECT-TYPE
```

```
        SYNTAX          Unsigned32 (0..65471)
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "The Target Destination Rbridge NickName Field as
            defined in RFC 6325 section 3.7 to be transmitted."
        REFERENCE "TRILL-FM and RFC6325 section 3.7"
        ::= { trillOamMepEntry 10 }

    trillOamMepTxLbmHC OBJECT-TYPE
        SYNTAX          Unsigned32(1..63)
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "The Hop Count to be transmitted.
            "
        REFERENCE "TRILL-FM section 9 and 3"
        ::= { trillOamMepEntry 11 }

    trillOamMepTxLbmReplyModeOob OBJECT-TYPE
        SYNTAX          TruthValue
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "True Indicates that Reply of Lbm is out of band and
            out of band IP Address TLV is to be transmitted.
            False indicates that In band reply is transmitted."
        REFERENCE "TRILL-FM 9.2.1"
        ::= { trillOamMepEntry 12 }

    trillOamMepTransmitLbmReplyIp OBJECT-TYPE
        SYNTAX          OCTET STRING
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "IP address for out of band IP Address TLV is to be transmitted."
        REFERENCE "TRILL-FM section 3"
        ::= { trillOamMepEntry 13 }

    trillOamMepTxLbmFlowEntropy OBJECT-TYPE
        SYNTAX          OCTET STRING
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "128 Byte Flow Entropy as defined in TRILL-FM to be transmitted."
        REFERENCE "TRILL-FM section 3"
        ::= { trillOamMepEntry 14 }
```

```
   trillOamMepTxPtmDestRName OBJECT-TYPE
       SYNTAX          Unsigned32 (0..65471)
       MAX-ACCESS      read-create
       STATUS          current
       DESCRIPTION
           "The Target Destination Rbridge NickName Field
           as defined in RFC 6325 section 3.7 to be transmitted."
       REFERENCE "TRILL-FM and RFC6325 section 3.7"
       ::= { trillOamMepEntry 15 }

   trillOamMepTxPtmHC OBJECT-TYPE
       SYNTAX          Unsigned32 (1..63)
       MAX-ACCESS      read-create
       STATUS          current
       DESCRIPTION
           "The Hop Count field to be transmitted.
           "
       REFERENCE "TRILL-FM section 3"
       ::= { trillOamMepEntry 16 }

   trillOamMepTxPtmReplyModeOob OBJECT-TYPE
       SYNTAX          TruthValue
       MAX-ACCESS      read-create
       STATUS          current
       DESCRIPTION
           "True Indicates that Reply of Ptm is out of band and
           out of band IP Address TLV is to be transmitted.
           False indicates that In band reply is transmitted."
       REFERENCE "TRILL-FM section 10"
       DEFVAL          { false }
       ::= { trillOamMepEntry 17 }

   trillOamMepTransmitPtmReplyIp OBJECT-TYPE
       SYNTAX          OCTET STRING
       MAX-ACCESS      read-create
       STATUS          current
       DESCRIPTION
           "IP address for out of band IP Address TLV is to be transmitted."
       REFERENCE "TRILL-FM section 3 and 10"
       ::= { trillOamMepEntry 18 }

   trillOamMepTxPtmFlowEntropy OBJECT-TYPE
       SYNTAX          OCTET STRING
       MAX-ACCESS      read-create
       STATUS          current
       DESCRIPTION
           "128 Byte Flow Entropy as defined in TRILL-FM to be transmitted."
       REFERENCE "TRILL-FM section 3"
```

```
        ::= { trillOamMepEntry 19 }

    trillOamMepTxPtmStatus OBJECT-TYPE
        SYNTAX          TruthValue
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "A Boolean flag set to true by the MEP Path Trace Initiator State
            Machine or an MIB manager to indicate that another Ptm is being
            transmitted.
            Reset to false by the MEP Initiator State Machine."
        REFERENCE "TRILL-FM section 10"
        DEFVAL          { false }
        ::= { trillOamMepEntry 20 }

    trillOamMepTxPtmResultOK OBJECT-TYPE
        SYNTAX          TruthValue
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "Indicates the result of the operation:
            - true  The Path Trace Message(s) will be (or has been) sent.
            - false The Path Trace Message(s) will not be sent."
        REFERENCE "TRILL-FM section 10"
        DEFVAL          { true }
        ::= { trillOamMepEntry 21 }

    trillOamMepTxPtmMessages OBJECT-TYPE
        SYNTAX          Integer32 (1..1024)
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "The number of Path Trace messages to be transmitted."
        REFERENCE "TRILL-FM section 10"
        ::= { trillOamMepEntry 22 }

    trillOamMepTxPtmSeqNumber OBJECT-TYPE
        SYNTAX          Unsigned32
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "The Path Trace Transaction Identifier of the first PTM (to be)
            sent. The value returned is undefined if
            trillOamMepTxPtmResultOK is false."
        REFERENCE "TRILL-FM section 10"
        ::= { trillOamMepEntry 23 }

    trillOamMepTxMtmTree OBJECT-TYPE
```

```
        SYNTAX          Unsigned32
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "The Multi-destination Tree is identifier for tree as defined in
            RFC6325."
        ::= { trillOamMepEntry 24 }

    trillOamMepTxMtmHC OBJECT-TYPE
        SYNTAX          Unsigned32(1..63)
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "The Hop Count field to be transmitted.
            "
        REFERENCE "TRILL-FM section 3, RFC 6325 section 3"
        ::= { trillOamMepEntry 25 }

    trillOamMepTxMtmReplyModeOob OBJECT-TYPE
        SYNTAX          TruthValue
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "True Indicates that Reply of Mtm is out of band and
            out of band IP Address TLV is to be transmitted.
            False indicates that In band reply is transmitted."
        REFERENCE "TRILL-FM section 11"
        ::= { trillOamMepEntry 26 }

    trillOamMepTransmitMtmReplyIp OBJECT-TYPE
        SYNTAX          OCTET STRING
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "IP address for out of band IP Address TLV is to be transmitted."
        REFERENCE "TRILL-FM section 11"
        ::= { trillOamMepEntry 27 }

    trillOamMepTxMtmFlowEntropy OBJECT-TYPE
        SYNTAX          OCTET STRING
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "128 Byte Flow Entropy as defined in TRILL-FM to be transmitted."
        REFERENCE "TRILL-FM section 3"
        ::= { trillOamMepEntry 28 }

    trillOamMepTxMtmStatus OBJECT-TYPE
```

```
        SYNTAX          TruthValue
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "A Boolean flag set to true by the MEP Multi Destination Initiator St
ate
            Machine or an MIB manager to indicate that another Mtm is being
            transmitted.
            Reset to false by the MEP Initiator State Machine."
        REFERENCE "TRILL-FM section 11"
        DEFVAL          { false }
        ::= { trillOamMepEntry 29 }

    trillOamMepTxMtmResultOK OBJECT-TYPE
        SYNTAX          TruthValue
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "Indicates the result of the operation:
            - true  The Multi-destination Message(s) will be (or has been) sent.
            - false The Multi-destination Message(s) will not be sent."
        REFERENCE "TRILL-FM section 11"
        DEFVAL          { true }
        ::= { trillOamMepEntry 30 }

    trillOamMepTxMtmMessages OBJECT-TYPE
        SYNTAX          Integer32 (1..1024)
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "The number of Multi Destination messages to be transmitted."
        REFERENCE "TRILL-FM section 11"
        ::= { trillOamMepEntry 31 }

    trillOamMepTxMtmSeqNumber OBJECT-TYPE
        SYNTAX          Unsigned32
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "The Multi-destination Transaction Identifier of the first MTM (to be
)
            sent. The value returned is undefined if
            trillOamMepTxMtmResultOK is false."
        REFERENCE "TRILL-FM section 11"
        ::= { trillOamMepEntry 32 }

    trillOamMepTxMtmScopeList OBJECT-TYPE
        SYNTAX          OCTET STRING
        MAX-ACCESS      read-create
        STATUS          current
```

```
        DESCRIPTION
            "The Multi-destination Rbridge Scope list, 2 OCTET per Rbridge."
        REFERENCE "TRILL-FM section 11"
        ::= { trillOamMepEntry 33 }


    -- ********************************************************************
    -- TRILL OAM Tx Measurement Configuration Table
    -- ********************************************************************

    trillOamMepFlowCfgTable OBJECT-TYPE
        SYNTAX          SEQUENCE OF TrillOamMepFlowCfgEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION
            "This table includes configuration objects and operations for
            the Trill OAM [TRILL-FM].

            Each row in the table represents a Flow configuration Entry for
            the defined MEP. This table uses four indices. The first
            three indices are the indices of the Maintenance Domain,
            MaNet, and MEP tables. The fourth index is the specific Flow
            configuration Entry on the selected MEP.

            Some writable objects in this table are only applicable in
            certain cases (as described under each object), and attempts to
            write values for them in other cases will be ignored."
        REFERENCE       "[TRILL-FM]"
        ::= { trillOamMep 2 }

    trillOamMepFlowCfgEntry OBJECT-TYPE
        SYNTAX          TrillOamMepFlowCfgEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION
            "The conceptual row of trillOamMepFlowCfgTable."
        INDEX           {
                            dot1agCfmMdIndex,
                            dot1agCfmMaIndex,
                            dot1agCfmMepIdentifier,
                            trillOamMepFlowCfgIndex
                        }
        ::= { trillOamMepFlowCfgTable 1 }

    TrillOamMepFlowCfgEntry ::= SEQUENCE {
            trillOamMepFlowCfgIndex       Unsigned32,
            trillOamMepFlowCfgFlowEntropy OCTET STRING,
            trillOamMepFlowCfgDestRName   Unsigned32,
```

```
            trillOamMepFlowCfgFlowHC      Unsigned32,
            trillOamMepFlowCfgRowStatus   RowStatus
            }

    trillOamMepFlowCfgIndex OBJECT-TYPE
        SYNTAX          Unsigned32 (1..65535)
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION
            "An index to the Trill OAM Mep Flow Configuration table which
            indicates the specific Flow for the MEP.

            The index is never reused for other flow sessions on the same
            MEP while this session is active.  The index value keeps
            increasing until it wraps to 0.
            This value can also be used in Flow-identifier TLV [TRILL-FM]"
        REFERENCE "TRILL-FM"
        ::= { trillOamMepFlowCfgEntry 1 }

    trillOamMepFlowCfgFlowEntropy OBJECT-TYPE
        SYNTAX          OCTET STRING
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "This is 128 byte of Flow Entropy as described in
            TRILL OAM [TRILL-FM]."
        REFERENCE "TRILL-FM section 3"
        ::= { trillOamMepFlowCfgEntry 2 }

    trillOamMepFlowCfgDestRName OBJECT-TYPE
        SYNTAX          Unsigned32 (0..65471)
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "The Target Destination Rbridge NickName Field as
            defined in RFC 6325 section 3.7 to be transmitted."
        REFERENCE "TRILL-FM section 3 and RFC 6325 section 3.7"
        ::= { trillOamMepFlowCfgEntry 3 }

    trillOamMepFlowCfgFlowHC OBJECT-TYPE
        SYNTAX          Unsigned32
        MAX-ACCESS      read-create
        STATUS          current
        DESCRIPTION
            "The Time to Live field to be transmitted.
            to be transmitted."
        REFERENCE "TRILL-FM section 3 and RFC 6325 section 3.7"
        ::= { trillOamMepFlowCfgEntry 4 }
```

```
   trillOamMepFlowCfgRowStatus OBJECT-TYPE
       SYNTAX          RowStatus
       MAX-ACCESS      read-create
       STATUS          current
       DESCRIPTION
           "The status of the row.

           The writable columns in a row cannot be changed if the row
           is active. All columns MUST have a valid value before a row
           can be activated."
       ::= { trillOamMepFlowCfgEntry 5 }


   -- ********************************************************************
   -- TRILL OAM Path Trace Reply Table
   -- ********************************************************************

   trillOamPtrTable OBJECT-TYPE
       SYNTAX          SEQUENCE OF TrillOamPtrEntry
       MAX-ACCESS      not-accessible
       STATUS          current
       DESCRIPTION
           "This table includes Path Trace Reply objects and operations for
           the Trill OAM [TRILL-FM].

           Each row in the table represents a Path Trace Reply Entry for
           the defined MEP and Transaction. This table uses four indices.
           The first three indices are the indices of the Maintenance Domain,
           MaNet, and MEP tables. The fourth index is the specific
           Transaction Identifier on the selected MEP.

           Some writable objects in this table are only applicable in
           certain cases (as described under each object), and attempts to
           write values for them in other cases will be ignored."
       REFERENCE       "TRILL-FM"
       ::= { trillOamMep 3 }

   trillOamPtrEntry OBJECT-TYPE
       SYNTAX          TrillOamPtrEntry
       MAX-ACCESS      not-accessible
       STATUS          current
       DESCRIPTION
           "The conceptual row of trillOamPtrTable."
       INDEX           {
                           dot1agCfmMdIndex,
                           dot1agCfmMaIndex,
                           dot1agCfmMepIdentifier,
                           trillOamMepPtrTransactionId
```

```
                        }
        ::= { trillOamPtrTable 1 }

   TrillOamPtrEntry ::= SEQUENCE {
            trillOamMepPtrTransactionId           Unsigned32,
            trillOamMepPtrHC                      Unsigned32,
            trillOamMepPtrFlag                    Unsigned32,
            trillOamMepPtrErrorCode               Unsigned32,
            trillOamMepPtrTerminalMep             TruthValue,
            trillOamMepPtrLastEgressId            Unsigned32,
            trillOamMepPtrIngress                 Dot1agCfmIngressActionFieldValu
e,
            trillOamMepPtrIngressMac              MacAddress,
            trillOamMepPtrIngressPortIdSubtype    LldpPortId,
            trillOamMepPtrIngressPortId           LldpPortId,
            trillOamMepPtrEgress                  Dot1agCfmEgressActionFieldValue
,
            trillOamMepPtrEgressMac               MacAddress,
            trillOamMepPtrEgressPortIdSubtype     LldpPortId,
            trillOamMepPtrEgressPortId            LldpPortId,
            trillOamMepPtrChassisIdSubtype        LldpChassisIdSubtype,
            trillOamMepPtrChassisId               LldpChassisId,
            trillOamMepPtrOrganizationSpecificTlv OCTET STRING,
            trillOamMepPtrNextHopNicknames        OCTET STRING
   }

   trillOamMepPtrTransactionId OBJECT-TYPE
       SYNTAX          Unsigned32 (0..4294967295)
       MAX-ACCESS      not-accessible
       STATUS          current
       DESCRIPTION
           "Transaction identifier/sequence number returned by a previous
           transmit path trace message command, indicating which PTM's
           response is going to be returned."
       REFERENCE       "TRILL-FM section 10"
       ::= { trillOamPtrEntry 1 }

   trillOamMepPtrHC  OBJECT-TYPE
       SYNTAX          Unsigned32 (1..63)
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "Hop Count field value for a returned PTR."
       REFERENCE       "TRILL-FM"
       ::= { trillOamPtrEntry 2 }

   trillOamMepPtrFlag OBJECT-TYPE
       SYNTAX          Unsigned32 (0..15)
       MAX-ACCESS      read-only
       STATUS          current
```

```
        DESCRIPTION
            "FCOI (TRILL OAM Message TLV) field value for a
             returned PTR."
        REFERENCE       "TRILL-FM, 9.4.2.1"
        ::= { trillOamPtrEntry 3 }

    trillOamMepPtrErrorCode OBJECT-TYPE
        SYNTAX          Unsigned32 (0..65535)
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Return Code and Return Sub code value for a returned PTR."
        REFERENCE       "TRILL-FM, 9.4.2.1"
        ::= { trillOamPtrEntry 4 }

    trillOamMepPtrTerminalMep OBJECT-TYPE
        SYNTAX          TruthValue
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "A boolean value stating whether the forwarded PTM reached a
            MEP enclosing its MA, as returned in the Terminal MEP flag of
            the Flags field."
        REFERENCE       "TRILL-FM"
        ::= { trillOamPtrEntry 5 }

    trillOamMepPtrLastEgressId OBJECT-TYPE
        SYNTAX          Unsigned32 (0..65535)
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "An Integer field holding the Last Egress Identifier returned
            in the PTR Upstream Rbridge nickname TLV of the PTR.
            The Last Egress Identifier identifies the Upstream Nickname"
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamPtrEntry 6 }

    trillOamMepPtrIngress OBJECT-TYPE
        SYNTAX          Dot1agCfmIngressActionFieldValue
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "The value returned in the Ingress Action Field of the PTM.
            The value ingNoTlv(0) indicates that no Reply Ingress TLV was
            returned in the PTM."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamPtrEntry 7 }
```

```
   trillOamMepPtrIngressMac OBJECT-TYPE
       SYNTAX          MacAddress
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "MAC address returned in the ingress MAC address field."
       REFERENCE       "TRILL-FM 8.4.1"
       ::= { trillOamPtrEntry 8 }

   trillOamMepPtrIngressPortIdSubtype OBJECT-TYPE
       SYNTAX          LldpPortId
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "Ingress Port ID. The format of this object is determined by
           the value of the trillOamMepPtrIngressPortIdSubtype object."
       REFERENCE       "TRILL-FM 8.4.1"
       ::= { trillOamPtrEntry 9 }

   trillOamMepPtrIngressPortId OBJECT-TYPE
       SYNTAX          LldpPortId
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "Ingress Port ID. The format of this object is determined by
           the value of the trillOamMepPtrIngressPortId object."
       REFERENCE       "TRILL-FM 8.4.1"
       ::= { trillOamPtrEntry 10 }

   trillOamMepPtrEgress OBJECT-TYPE
       SYNTAX          Dot1agCfmEgressActionFieldValue
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "The value returned in the Egress Action Field of the PTM.
           The value ingNoTlv(0) indicates that no Reply Egress TLV was
           returned in the PTM."
       REFERENCE       "TRILL-FM 8.4.1"
       ::= { trillOamPtrEntry 11 }

   trillOamMepPtrEgressMac OBJECT-TYPE
       SYNTAX          MacAddress
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "MAC address returned in the egress MAC address field."
       REFERENCE       "TRILL-FM 8.4.1"
       ::= { trillOamPtrEntry 12 }
```

```
   trillOamMepPtrEgressPortIdSubtype OBJECT-TYPE
       SYNTAX          LldpPortId
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "Egress Port ID. The format of this object is determined by
           the value of the trillOamMepPtrEgressPortIdSubtype object."
       REFERENCE       "TRILL-FM 8.4.1"
       ::= { trillOamPtrEntry 13 }

   trillOamMepPtrEgressPortId OBJECT-TYPE
       SYNTAX          LldpPortId
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "Egress Port ID. The format of this object is determined by
           the value of the trillOamMepPtrEgressPortId object."
       REFERENCE       "TRILL-FM 8.4.1"
       ::= { trillOamPtrEntry 14 }

   trillOamMepPtrChassisIdSubtype OBJECT-TYPE
       SYNTAX          LldpChassisIdSubtype
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "This object specifies the format of the Chassis ID returned
           in the Sender ID TLV of the PTR, if any.  This value is
           meaningless if the trillOamMepPtrChassisId has a length of 0."
       REFERENCE       "TRILL-FM 8.4.1"
       ::= { trillOamPtrEntry 15 }

   trillOamMepPtrChassisId OBJECT-TYPE
       SYNTAX          LldpChassisId
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "The Chassis ID returned in the Sender ID TLV of the PTR, if
           any. The format of this object is determined by the
           value of the trillOamMepPtrChassisIdSubtype object."
       REFERENCE       "TRILL-FM 8.4.1"
       ::= { trillOamPtrEntry 16 }

   trillOamMepPtrOrganizationSpecificTlv OBJECT-TYPE
       SYNTAX          OCTET STRING (SIZE  (0..0 | 4..1500))
       MAX-ACCESS      read-only
       STATUS          current
       DESCRIPTION
           "All Organization specific TLVs returned in the PTR, if
```

```
            any.  Includes all octets including and following the TLV
            Length field of each TLV, concatenated together."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamPtrEntry 17 }

    trillOamMepPtrNextHopNicknames OBJECT-TYPE
        SYNTAX          OCTET STRING (SIZE  (0..0 | 4..1500))
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Next hop Rbridge List TLV returned in the PTR, if
            any.  Includes all octets including and following the TLV
            Length field of each TLV, concatenated together."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamPtrEntry 18 }


    -- ********************************************************************
    -- TRILL OAM Multi Destination Reply Table
    -- ********************************************************************

    trillOamMtrTable OBJECT-TYPE
        SYNTAX          SEQUENCE OF TrillOamMtrEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION
            "This table includes Multi-destination Reply objects and
             operations for the Trill OAM [TRILL-FM].

            Each row in the table represents a Multi-destination Reply
            Entry for the defined MEP and Transaction.
            This table uses five indices.
            The first three indices are the indices of the Maintenance Domain,
            MaNet, and MEP tables. The fourth index is the specific
            Transaction Identifier on the selected MEP.
            The fifth index is the receive order of Multi-destination
            replies.

            Some writable objects in this table are only applicable in
            certain cases (as described under each object), and attempts to
            write values for them in other cases will be ignored."
        REFERENCE       "TRILL-FM"
        ::= { trillOamMep 4 }

    trillOamMtrEntry OBJECT-TYPE
        SYNTAX          TrillOamMtrEntry
        MAX-ACCESS      not-accessible
        STATUS          current
```

```
        DESCRIPTION
            "The conceptual row of trillOamMtrTable."
        INDEX           {
                            dot1agCfmMdIndex,
                            dot1agCfmMaIndex,
                            dot1agCfmMepIdentifier,
                            trillOamMepPtrTransactionId,
                            trillOamMepMtrReceiveOrder
                        }
        ::= { trillOamMtrTable 1 }

    TrillOamMtrEntry ::= SEQUENCE {
            trillOamMepMtrTransactionId           Unsigned32,
            trillOamMepMtrReceiveOrder            Unsigned32,
            trillOamMepMtrFlag                    Unsigned32,
            trillOamMepMtrErrorCode               Unsigned32,
            trillOamMepMtrLastEgressId            Unsigned32,
            trillOamMepMtrIngress                 Dot1agCfmIngressActionFieldValu
e,
            trillOamMepMtrIngressMac              MacAddress,
            trillOamMepMtrIngressPortIdSubtype    LldpPortId,
            trillOamMepMtrIngressPortId           LldpPortId,
            trillOamMepMtrEgress                  Dot1agCfmEgressActionFieldValue
,
            trillOamMepMtrEgressMac               MacAddress,
            trillOamMepMtrEgressPortIdSubtype     LldpPortId,
            trillOamMepMtrEgressPortId            LldpPortId,
            trillOamMepMtrChassisIdSubtype        LldpChassisIdSubtype,
            trillOamMepMtrChassisId               LldpChassisId,
            trillOamMepMtrOrganizationSpecificTlv OCTET STRING,
            trillOamMepMtrNextHopNicknames        OCTET STRING,
            trillOamMepMtrReceiverAvailability    TruthValue,
            trillOamMepMtrReceiverCount           TruthValue
    }

    trillOamMepMtrTransactionId OBJECT-TYPE
        SYNTAX          Unsigned32 (0..4294967295)
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION
            "Transaction identifier/sequence number returned by a previous
            transmit Multi-destination message command, indicating
            which MTM's response is going to be returned."
        REFERENCE       "TRILL-FM section 11"
        ::= { trillOamMtrEntry 1 }

    trillOamMepMtrReceiveOrder OBJECT-TYPE
        SYNTAX          Unsigned32 (1..4294967295)
        MAX-ACCESS      not-accessible
        STATUS          current
```

```
        DESCRIPTION
            "An index to distinguish among multiple MTR with same
             MTR Transaction Identifier field value.
             trillOamMepMtrReceiveOrder are assigned sequentially from 1,
             in the order that the Multi-destination Tree Initiator
             received the MTRs."
        REFERENCE   "TRILL-FM section 11"
        ::= { trillOamMtrEntry 2 }

    trillOamMepMtrFlag OBJECT-TYPE
        SYNTAX          Unsigned32 (0..15)
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "FCOI (TRILL OAM Message TLV) field value for a
             returned MTR."
        REFERENCE       "TRILL-FM, 8.4.2"
        ::= { trillOamMtrEntry 3 }

    trillOamMepMtrErrorCode OBJECT-TYPE
        SYNTAX          Unsigned32 (0..65535)
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Return Code and Return Sub code value for a returned MTR."
        REFERENCE       "TRILL-FM, 8.4.2"
        ::= { trillOamMtrEntry 4 }

    trillOamMepMtrLastEgressId OBJECT-TYPE
        SYNTAX          Unsigned32 (0..65535)
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "An Integer field holding the Last Egress Identifier returned
            in the MTR Upstream Rbridge Nickname TLV of the MTR.
            The Last Egress Identifier identifies the Upstream Nickname."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 5 }

    trillOamMepMtrIngress OBJECT-TYPE
        SYNTAX          Dot1agCfmIngressActionFieldValue
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "The value returned in the Ingress Action Field of the MTR.
            The value ingNoTlv(0) indicates that no Reply Ingress TLV was
            returned in the MTM."
        REFERENCE       "TRILL-FM 11.2.3"
```

```
        ::= { trillOamMtrEntry 6 }

    trillOamMepMtrIngressMac OBJECT-TYPE
        SYNTAX          MacAddress
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "MAC address returned in the ingress MAC address field."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 7 }

    trillOamMepMtrIngressPortIdSubtype OBJECT-TYPE
        SYNTAX          LldpPortId
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Ingress Port ID. The format of this object is determined by
            the value of the trillOamMepMtrIngressPortIdSubtype object."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 8 }

    trillOamMepMtrIngressPortId OBJECT-TYPE
        SYNTAX          LldpPortId
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Ingress Port ID. The format of this object is determined by
            the value of the trillOamMepMtrIngressPortId object."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 9 }

    trillOamMepMtrEgress OBJECT-TYPE
        SYNTAX          Dot1agCfmEgressActionFieldValue
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "The value returned in the Egress Action Field of the MTR.
            The value ingNoTlv(0) indicates that no Reply Egress TLV was
            returned in the MTR."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 10 }

    trillOamMepMtrEgressMac OBJECT-TYPE
        SYNTAX          MacAddress
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "MAC address returned in the egress MAC address field."
```

```
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 11 }

    trillOamMepMtrEgressPortIdSubtype OBJECT-TYPE
        SYNTAX          LldpPortId
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Egress Port ID. The format of this object is determined by
            the value of the trillOamMepMtrEgressPortIdSubtype object."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 12 }

    trillOamMepMtrEgressPortId OBJECT-TYPE
        SYNTAX          LldpPortId
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Egress Port ID. The format of this object is determined by
            the value of the trillOamMepMtrEgressPortId object."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 13 }

    trillOamMepMtrChassisIdSubtype OBJECT-TYPE
        SYNTAX          LldpChassisIdSubtype
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "This object specifies the format of the Chassis ID returned
            in the Sender ID TLV of the MTR, if any.  This value is
            meaningless if the trillOamMepMtrChassisId has a length of 0."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 14 }

    trillOamMepMtrChassisId OBJECT-TYPE
        SYNTAX          LldpChassisId
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "The Chassis ID returned in the Sender ID TLV of the MTR, if
            any. The format of this object is determined by the
            value of the trillOamMepMtrChassisIdSubtype object."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 15 }

    trillOamMepMtrOrganizationSpecificTlv OBJECT-TYPE
        SYNTAX          OCTET STRING (SIZE  (0..0 | 4..1500))
        MAX-ACCESS      read-only
```

```
        STATUS          current
        DESCRIPTION
            "All Organization specific TLVs returned in the MTR, if
            any.  Includes all octets including and following the TLV
            Length field of each TLV, concatenated together."
        REFERENCE       "TRILL-FM 8.4.1"
        ::= { trillOamMtrEntry 16 }

    trillOamMepMtrNextHopNicknames OBJECT-TYPE
        SYNTAX          OCTET STRING (SIZE  (0..0 | 4..1500))
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Next hop Rbridge List TLV returned in the PTR, if
            any.  Includes all octets including and following the TLV
            Length field of each TLV, concatenated together."
        REFERENCE       "TRILL-FM 8.4.3"
        ::= { trillOamMtrEntry 17 }

    trillOamMepMtrReceiverAvailability OBJECT-TYPE
        SYNTAX          TruthValue
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "True value indicates that MTR response contained
            Multicast receiver availability TLV"
        REFERENCE       "TRILL-FM 8.4.10"
        ::= { trillOamMtrEntry 18 }

    trillOamMepMtrReceiverCount OBJECT-TYPE
        SYNTAX          TruthValue
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
            "Indicates the number of Multicast receivers available on
            responding RBridge on the VLAN specified by the
            diagnostic VLAN."
        REFERENCE       "TRILL-FM 8.4.10"
        ::= { trillOamMtrEntry 19 }

    -- ********************************************************************
    -- TRILL OAM MEP Database Table
    -- ********************************************************************

    trillOamMepDbTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF TrillOamMepDbEntry
        MAX-ACCESS  not-accessible
        STATUS      current
```

```
        DESCRIPTION
           "This table is an extension of the dot1agCfmMepDbTable and rows
            are automatically added or deleted from this table based upon
            row creation and destruction of the dot1agCfmMepDbTable.
           "
        REFERENCE
           "[TRILL-FM]"
        ::= { trillOamMep 5 }

    trillOamMepDbEntry OBJECT-TYPE
        SYNTAX      TrillOamMepDbEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "The conceptual row of trillOamMepDbTable."
        AUGMENTS {
                 dot1agCfmMepDbEntry
                 }
        ::= { trillOamMepDbTable 1 }

    TrillOamMepDbEntry ::= SEQUENCE {
          trillOamMepDbFlowIndex         Unsigned32,
          trillOamMepDbFlowEntropy       OCTET STRING,
          trillOamMepDbFlowState         Dot1agCfmRemoteMepState,
          trillOamMepDbFlowFailedOkTime  TimeStamp,
          trillOamMepDbRbridgeName       Unsigned32,
          trillOamMepDbLastGoodSeqNum    Counter32
        }

    trillOamMepDbFlowIndex OBJECT-TYPE
        SYNTAX        Unsigned32 (1..65535)
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
          "This object identifies the Flow. If Flow Identifier TLV is received
           than index received can also be used.
          "
        REFERENCE "TRILL-FM"
        ::= {trillOamMepDbEntry 1 }

    trillOamMepDbFlowEntropy OBJECT-TYPE
        SYNTAX        OCTET STRING
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
          "128 byte Flow Entropy.
          "
        REFERENCE "TRILL-FM section 3."
```

```
       ::= {trillOamMepDbEntry 2 }

   trillOamMepDbFlowState OBJECT-TYPE
       SYNTAX       Dot1agCfmRemoteMepState
       MAX-ACCESS    read-only
       STATUS        current
       DESCRIPTION
           "The operational state of the remote MEP (flow based)
            IFF State machines. State Machine is running now per
            flow."
       REFERENCE "TRILL-FM"
       ::= {trillOamMepDbEntry 3 }

    trillOamMepDbFlowFailedOkTime OBJECT-TYPE
       SYNTAX        TimeStamp
       MAX-ACCESS    read-only
       STATUS        current
       DESCRIPTION
           "The Time (sysUpTime) at which the Remote Mep Flow state
            machine last entered either the RMEP_FAILED or RMEP_OK
            state.
           "
       REFERENCE "TRILL-FM"
       ::= {trillOamMepDbEntry 4 }

   trillOamMepDbRbridgeName OBJECT-TYPE
       SYNTAX        Unsigned32(0..65471)
       MAX-ACCESS    read-only
       STATUS        current
       DESCRIPTION
            "Remote MEP Rbridge Nickname"
       REFERENCE "TRILL-FM RFC 6325 section 3"
       ::= {trillOamMepDbEntry 5 }

   trillOamMepDbLastGoodSeqNum OBJECT-TYPE
       SYNTAX        Counter32
       MAX-ACCESS    read-only
       STATUS        current
       DESCRIPTION
            "Last Sequence Number received."
       REFERENCE "TRILL-FM 13.1"
       ::= {trillOamMepDbEntry 6}

   -- ********************************************************************
***
   -- TRILL OAM MIB NOTIFICATIONS (TRAPS)
   -- This notification is sent to management entity whenever a MEP loses/restor
es
   -- contact with its peer Flow Meps
   -- ********************************************************************
***
```

```
   trillOamFaultAlarm NOTIFICATION-TYPE
       OBJECTS         { trillOamMepDbFlowState }
       STATUS          current
       DESCRIPTION
           "A MEP Flow has a persistent defect condition.
            A notification (fault alarm) is sent to the management
            entity with the OID of the Flow that has detected the fault.

           The management entity receiving the notification can identify
           the system from the network source address of the
           notification, and can identify the Flow reporting the defect
           by the indices in the OID of the
           trillOamMepFlowIndex, and trillOamFlowDefect
           variable in the notification:

              dot1agCfmMdIndex - Also the index of the MEP's
                                 Maintenance Domain table entry
                                 (dot1agCfmMdTable).
              dot1agCfmMaIndex - Also an index (with the MD table index)
                                 of the MEP's Maintenance Association
                                 network table entry
                                 (dot1agCfmMaNetTable), and (with the MD
                                 table index and component ID) of the
                                 MEP's MA component table entry
                                 (dot1agCfmMaCompTable).
              dot1agCfmMepIdentifier - MEP Identifier and final index
                                 into the MEP table (dot1agCfmMepTable).
              trillOamMepFlowCfgIndex - Index identifies
                                indicates the specific Flow for the MEP"
       REFERENCE       "TRILL-FM"
      ::= { trillOamNotifications 1 }

   -- ************************************************************************
***
   -- TRILL OAM MIB Module - Conformance Information
   -- ************************************************************************
***

   trillOamMibCompliances  OBJECT IDENTIFIER
       ::= { trillOamMibConformance 1 }

   trillOamMibGroups  OBJECT IDENTIFIER
       ::= { trillOamMibConformance 2 }


   -- ****************************************************************
   -- TRILL OAM MIB Units of conformance
   -- ****************************************************************

   trillOamMepMandatoryGroup OBJECT-GROUP
```

```
        OBJECTS         {
                            trillOamMepRName,
                            trillOamMepNextPtmTId,
                            trillOamMepNextMtmTId,
                            trillOamMepPtrIn,
                            trillOamMepPtrInOutofOrder,
                            trillOamMepPtrOut,
                            trillOamMepMtrIn,
                            trillOamMepMtrInOutofOrder,
                            trillOamMepMtrOut,
                            trillOamMepTxLbmDestRName,
                            trillOamMepTxLbmHC,
                            trillOamMepTxLbmReplyModeOob,
                            trillOamMepTransmitLbmReplyIp,
                            trillOamMepTxLbmFlowEntropy,
                            trillOamMepTxPtmDestRName,
                            trillOamMepTxPtmHC,
                            trillOamMepTxPtmReplyModeOob,
                            trillOamMepTransmitPtmReplyIp,
                            trillOamMepTxPtmFlowEntropy,
                            trillOamMepTxPtmStatus,
                            trillOamMepTxPtmResultOK,
                            trillOamMepTxPtmMessages,
                            trillOamMepTxPtmSeqNumber,
                            trillOamMepTxMtmTree,
                            trillOamMepTxMtmHC,
                            trillOamMepTxMtmReplyModeOob,
                            trillOamMepTransmitMtmReplyIp,
                            trillOamMepTxMtmFlowEntropy,
                            trillOamMepTxMtmStatus,
                            trillOamMepTxMtmResultOK,
                            trillOamMepTxMtmMessages,
                            trillOamMepTxMtmSeqNumber,
                            trillOamMepTxMtmScopeList
                        }
        STATUS          current
        DESCRIPTION
            "Mandatory objects for the TRILL OAM MEP group."
        ::= { trillOamMibGroups 1 }

    trillOamMepFlowCfgTableGroup OBJECT-GROUP
        OBJECTS         {
                            trillOamMepFlowCfgFlowEntropy,
                            trillOamMepFlowCfgDestRName,
                            trillOamMepFlowCfgFlowHC,
                            trillOamMepFlowCfgRowStatus
                        }
        STATUS          current
```

```
        DESCRIPTION
            "Trill OAM MEP Flow Configuration objects group."
        ::= { trillOamMibGroups 2 }

    trillOamPtrTableGroup OBJECT-GROUP
        OBJECTS         {
                            trillOamMepPtrHC,
                            trillOamMepPtrFlag,
                            trillOamMepPtrErrorCode,
                            trillOamMepPtrTerminalMep,
                            trillOamMepPtrLastEgressId,
                            trillOamMepPtrIngress,
                            trillOamMepPtrIngressMac,
                            trillOamMepPtrIngressPortIdSubtype,
                            trillOamMepPtrIngressPortId,
                            trillOamMepPtrEgress,
                            trillOamMepPtrEgressMac,
                            trillOamMepPtrEgressPortIdSubtype,
                            trillOamMepPtrEgressPortId,
                            trillOamMepPtrChassisIdSubtype,
                            trillOamMepPtrChassisId,
                            trillOamMepPtrOrganizationSpecificTlv,
                            trillOamMepPtrNextHopNicknames
                        }
        STATUS          current
        DESCRIPTION
            "Trill OAM MEP PTR objects group."
        ::= { trillOamMibGroups 3 }

    trillOamMtrTableGroup OBJECT-GROUP
        OBJECTS         {
                            trillOamMepMtrFlag,
                            trillOamMepMtrErrorCode,
                            trillOamMepMtrLastEgressId,
                            trillOamMepMtrIngress,
                            trillOamMepMtrIngressMac,
                            trillOamMepMtrIngressPortIdSubtype,
                            trillOamMepMtrIngressPortId,
                            trillOamMepMtrEgress,
                            trillOamMepMtrEgressMac,
                            trillOamMepMtrEgressPortIdSubtype,
                            trillOamMepMtrEgressPortId,
                            trillOamMepMtrChassisIdSubtype,
                            trillOamMepMtrChassisId,
                            trillOamMepMtrOrganizationSpecificTlv,
                            trillOamMepMtrNextHopNicknames,
                            trillOamMepMtrReceiverAvailability,
                            trillOamMepMtrReceiverCount
```

```
                    }
        STATUS          current
        DESCRIPTION
            "Trill OAM MEP MTR objects group."
        ::= { trillOamMibGroups 4 }

    trillOamMepDbGroup  OBJECT-GROUP
        OBJECTS   {
          trillOamMepDbFlowIndex,
          trillOamMepDbFlowEntropy,
          trillOamMepDbFlowState,
          trillOamMepDbFlowFailedOkTime,
          trillOamMepDbRbridgeName,
          trillOamMepDbLastGoodSeqNum
        }

        STATUS      current
        DESCRIPTION
           "Trill OAM MEP DB objects group."
        ::= { trillOamMibGroups 5 }

    trillOamNotificationGroup NOTIFICATION-GROUP
        NOTIFICATIONS  {
            trillOamFaultAlarm
        }
        STATUS current
        DESCRIPTION
            "Objects for Notification Group"
        ::= { trillOamMibGroups 6 }

    -- ****************************************************************
    -- TRILL OAM MIB Module Compliance statements
    -- ****************************************************************

    trillOamMibCompliance MODULE-COMPLIANCE
        STATUS          current
        DESCRIPTION
            "The compliance statement for the TRILL OAM MIB."
        MODULE          -- this module
        MANDATORY-GROUPS {
                            trillOamMepMandatoryGroup,
                            trillOamMepFlowCfgTableGroup,
                            trillOamPtrTableGroup,
                            trillOamMtrTableGroup,
                            trillOamMepDbGroup,
                            trillOamNotificationGroup
                        }
        ::= { trillOamMibCompliances 1 }
```

```
   -- Compliance requirement for read-only implementation.

   trillOamMibReadOnlyCompliance MODULE-COMPLIANCE
       STATUS current
       DESCRIPTION
           "Compliance requirement for implementation that only
            provide read-only support for TRILL-OAM-MIB.
            Such devices can be monitored but cannot be configured
            using this MIB module
           "
       MODULE -- this module
       MANDATORY-GROUPS {
                           trillOamMepMandatoryGroup,
                           trillOamMepFlowCfgTableGroup,
                           trillOamPtrTableGroup,
                           trillOamMtrTableGroup,
                           trillOamMepDbGroup,
                           trillOamNotificationGroup
                         }
       -- trillOamMepTable

       OBJECT trillOamMepTxLbmDestRName
       MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

       OBJECT trillOamMepTxLbmHC
       MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

       OBJECT trillOamMepTxLbmReplyModeOob
       MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

       OBJECT trillOamMepTransmitLbmReplyIp
       MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

       OBJECT trillOamMepTxLbmFlowEntropy
       MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

       OBJECT trillOamMepTxPtmDestRName
       MIN-ACCESS  read-only
```

```
            DESCRIPTION
                "Write access is not required."

        OBJECT trillOamMepTxPtmHC
        MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

        OBJECT trillOamMepTxPtmReplyModeOob
        MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

        OBJECT trillOamMepTransmitPtmReplyIp
        MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

        OBJECT trillOamMepTxPtmFlowEntropy
        MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

        OBJECT trillOamMepTxPtmStatus
        MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

        OBJECT trillOamMepTxPtmResultOK
        MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

        OBJECT trillOamMepTxPtmMessages
        MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

        OBJECT trillOamMepTxPtmSeqNumber
        MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."

        OBJECT trillOamMepTxMtmTree
        MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required."
```

```
        OBJECT trillOamMepTxMtmHC
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepTxMtmReplyModeOob
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepTransmitMtmReplyIp
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepTxMtmFlowEntropy
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepTxMtmStatus
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepTxMtmResultOK
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepTxMtmMessages
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepTxMtmSeqNumber
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepTxMtmScopeList
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."


    -- trillOamMepFlowCfgTable
```

```
        OBJECT trillOamMepFlowCfgFlowEntropy
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepFlowCfgDestRName
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepFlowCfgFlowHC
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."

        OBJECT trillOamMepFlowCfgRowStatus
        MIN-ACCESS  read-only
             DESCRIPTION
                 "Write access is not required."


        ::= { trillOamMibCompliances 2 }


    END
```

8. Security Considerations

This MIB relates to a system that will provide network connectivity and packet forwarding services. As such, improper manipulation of the objects represented by this MIB may result in denial of service to a large number of end-users.

There are number of management objects defined in this MIB module with a MAX-ACCESS clause of read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have negative effect on sensitivity/vulnerability are described below.

Some of the readable objects in this MIB module (objects with a MAC-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control GET and/or NOTIFY access to these objects and possibly to encrypt the values of these objects when sending them over the network via SNMP.

SNMP version prior to SNMPv3 did not include adequate security. Even

if the network itself is secure, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanism (for authentication and privacy).

Further, deployment of SNMP version prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

9. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER value recorded in the SMI Numbers registry:

```
Descriptor   OBJECT   IDENTIFIER  value
----------------------------------------
trillOamMIB  { mib-2 xxx }
```

Editor’s Note (to be removed prior to publication): the IANA is requested to assign a value for "xxx" under the ’mib-2’ subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.

10. References

10.1. Normative References

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

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD

58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

10.2. Informative References

[RFC6905] Senevirathne, T., Bond, D., Aldrin, S., Li, Y., and R. Watve, "Requirements for Operations, Administration, and Maintenance (OAM) in Transparent Interconnection of Lots of Links (TRILL)", RFC 6905, March 2013.

[TRILLOAMFM] Salam, S., et.al., "TRILL OAM Framework", draft-ietf-trill-oam-framework, Work in Progress,November, 2012.

[TRILL-FM] Senevirathne, T., et.al., "TRILL Fault Management", draft-tissa-trill-oam-fm, Work in Progress, Feburary, 2013.

11. Acknowledgments

We wish to thank members of the IETF TRILL WG for their comments and suggestions. Detailed comments were provided by Sam Aldrin, and Donald Eastlake.

Authors' Addresses

Deepak Kumar
Cisco
510 McCarthy Blvd,
Milpitas, CA 95035, USA
Phone : +1 408-853-9760
Email: dekumar@cisco.com

Samer Salam
Cisco
595 Burrard St. Suite 2123
Vancouver, BC V7X 1J1, Canada
Email: ssalam@cisco.com

Tissa Senevirathne
Cisco
375 East Tasman Drive
San Jose, CA 95134, USA
Email: tsenevir@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015

M. Wasserman
Painless Security
D. Eastlake
D. Zhang
Huawei Technologies

July 4, 2014

Transparent Interconnection of Lots of Links (TRILL) over IP
draft-ietf-trill-over-ip-01.txt

Abstract

The Transparent Interconnection of Lots of Links (TRILL) protocol is implemented by devices called TRILL Switches or RBridges (Routing Bridges). TRILL supports both point-to-point and multi-access links and is designed so that a variety of link protocols can be used between TRILL switch ports. This document standardizes methods for encapsulating TRILL in IP(v4 or v6) to provide a unified TRILL campus.

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 January 5, 2015.

Copyright Notice

Table of Contents

1. Requirements Terminology

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].

2. Introduction

TRILL switches (RBridges) are devices that implement the IETF TRILL protocol [RFC6325] [RFC7176] [RFC7177].

RBridges provide transparent forwarding of frames within an arbitrary network topology, using least cost paths for unicast traffic. They support not only VLANs and Fine Grained Labels [RFC7172] but also multipathing of unicast and multi-destination traffic. They use IS-IS link state routing and encapsulation with a hop count. They are compatible with IEEE 802.1 customer bridges, and can incrementally replace them.

Ports on different RBridges can communicate with each other over various link types, such as Ethernet [RFC6325], pseudowires [RFC7173], or PPP [RFC6361].

This document defines a method for RBridges to communicate over UDP/IP(v4 or v6). TRILL over IP will allow remote, Internet-connected RBridges to form a single RBridge campus, or multiple TRILL over IP networks within a campus to be connected as a single TRILL campus via a TRILL over IP backbone.

TRILL over IP connects RBridge ports using IPv4 or IPv6 as a transport in such a way that the ports appear to TRILL to be connected by a single multi-access link. Therefore, if more than two RBridge ports are connected via a single TRILL over IP link, any pair of them can communicate.

To support the scenarios where RBridges are connected via links (such as the public Internet) that are not under the same administrative control as the TRILL campus, this document specifies the use of Datagram Transport Layer Security (DTLS) [RFC6347] to secure the communications between RBridges running TRILL over IP.

3. Use Cases for TRILL over IP

This section introduces two application scenarios (a remote office scenario and an IP backbone scenario) which cover the most typical of situations where network administrators may choose to use TRILL over an IP network.

3.1. Remote Office Scenario

In the Remote Office Scenario, a remote TRILL network is connected to a TRILL campus across a multihop IP network, such as the public Internet. The TRILL network in the remote office becomes a logical part of TRILL campus, and nodes in the remote office can be attached

to the same VLANs or Fine Grained Labels[RFC7172] as local campus nodes. In many cases, a remote office may be attached to the TRILL campus by a single pair of RBridges, one on the campus end, and the other in the remote office. In this use case, the TRILL over IP link will often cross logical and physical IP networks that do not support TRILL, and are not under the same administrative control as the TRILL campus.

3.2. IP Backbone Scenario

In the IP Backbone Scenario, TRILL over IP is used to connect a number of TRILL networks to form a single TRILL campus. For example, a TRILL over IP backbone could be used to connect multiple TRILL networks on different floors of a large building, or to connect TRILL networks in separate buildings of a multi-building site. In this use case, there may often be several TRILL switches on a single TRILL over IP link, and the IP link(s) used by TRILL over IP are typically under the same administrative control as the rest of the TRILL campus.

3.3. Important Properties of the Scenarios

There are a number of differences between the above two application scenarios, some of which drive features of this specification. These differences are especially pertinent to the security requirements of the solution, how multicast data frames are handled, and how the TRILL switch ports discover each other.

3.3.1. Security Requirements

In the IP Backbone Scenario, TRILL over IP is used between a number of RBridge ports, on a network link that is in the same administrative control as the remainder of the TRILL campus. While it is desirable in this scenario to prevent the association of rogue RBridges, this can be accomplished using existing IS-IS security mechanisms. There may be no need to protect the data traffic, beyond any protections that are already in place on the local network.

In the Remote Office Scenario, TRILL over IP may run over a network that is not under the same administrative control as the TRILL network. Nodes on the network may think that they are sending traffic locally, while that traffic is actually being sent, in a UDP/IP tunnel, over the public Internet. It is necessary in this scenario to protect the integrity and confidentiality of user traffic, as well as ensuring that no unauthorized RBridges can gain access to the RBridge campus. The issues of protecting integrity and confidentiality of user traffic are addressed by using DTLS for both IS-IS frames and data frames between RBridges in this scenario.

3.3.2. Multicast Handling

In the IP Backbone scenario, native multicast may be supported on the TRILL over IP link. If so, it can be used to send TRILL IS-IS and multicast data packets, as discussed later in this document. Alternatively, multi-destination packets can be transmitted serially.

In the Remote Office Scenario there will often be only one pair of RBridges connecting a given site and, even when multiple RBridges are used to connect a Remote Office to the TRILL campus, the intervening network may not provide reliable (or any) multicast connectivity. Issues such as complex key management also makes it difficult to provide strong data integrity and confidentiality protections for multicast traffic. For all of these reasons, the connections between local and remote RBridges will be treated like point-to-point links, and all TRILL IS-IS control messages and multicast data packets that are transmitted between the Remote Office and the TRILL campus will be serially transmitted, as discussed later in this document.

3.3.3. RBridge Neighbor Discovery

In the IP Backbone Scenario, RBridges that use TRILL over IP will use the normal TRILL IS-IS Hello mechanisms to discover the existence of other RBridges on the link [RFC7177], and to establish authenticated communication with those RBridges.

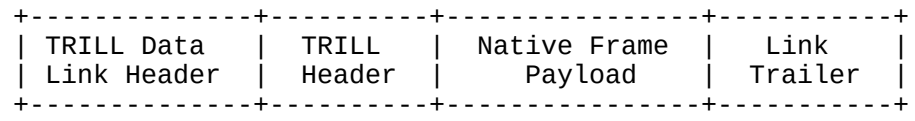
In the Remote Office Scenario, a DTLS session will need to be established between RBridges before TRILL IS-IS traffic can be exchanged, as discussed below. In this case, one of the RBridges will need to be configured to establish a DTLS session with the other RBridge. This will typically be accomplished by configuring the RBridge at a Remote Office to initiate a DTLS session, and subsequent TRILL exchanges, with a TRILL over IP-enabled RBridge attached to the TRILL campus.

4. TRILL Packet Formats

To support the TRILL base protocol standard [RFC6325]. , two types of packets will be transmitted between RBridges: TRILL Data frames and TRILL IS-IS packets.

4.1. TRILL Data Packet

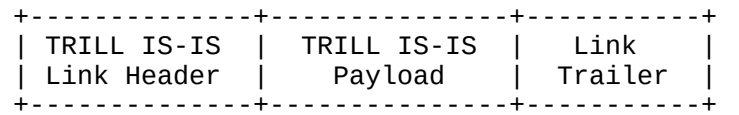
The on-the-wire form of a TRILL Data packet in transit between two neighboring RBridges is as shown below:



Where the Encapsulated Native Frame is similar to Ethernet frame format with a VLAN tag or Fine Grained Label [RFC7172] but with no trailing Frame Check Sequence (FCS).

4.2. TRILL IS-IS Packet

TRILL IS-IS packets are formatted on-the-wire as follows:



The Link Header and Link Trailer in these formats depend on the specific link technology. The Link Header usually contains one or more fields that distinguish TRILL Data from TRILL IS-IS. For example, over Ethernet, the TRILL Data Link Header ends with the TRILL Ethertype while the TRILL IS-IS Link Header ends with the L2-IS-IS Ethertype; on the other hand, over PPP, there are no Ethertypes but PPP protocol code points are included that distinguish TRILL Data from TRILL IS-IS.

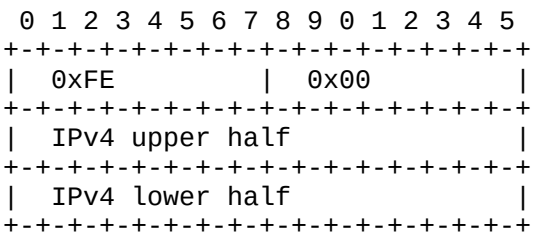
In TRILL over IP, we will use UDP/IP (v4 or v6) as the link header, and the TRILL packet type will be determined based on the UDP destination port number. In TRILL over IP, no Link Trailer is specified, although one may be added when the resulting IP packets are encapsulated for transmission on a network (e.g. Ethernet).

5. Link Protocol Specifics

TRILL Data packets can be unicast to a specific RBridge or multicast to all RBridges on the link. TRILL IS-IS packets are always multicast to all other RBridge on the link (except for MTU PDUs, which may be unicast). On Ethernet links, the Ethernet multicast address All-RBridges is used for TRILL Data and All-IS-IS-RBridges for TRILL IS-IS.

To properly handle TRILL base protocol packets on a TRILL over IP link, either native multicast mode must be enabled on that link, or multicast must be simulated using serial unicast, as discussed below.

In TRILL Hello PDUs used on TRILL IP links, the IP addresses of the connected IP ports are their real SNPA (SubNetwork Point of Attachment) addresses and, for IPv6, the 16-byte IPv6 address is used; however, for easy of code re-use designed for common 48-bit SNPAs, for TRILL over IPv4, a 48-bit synthetic SNPA that looks like a unicast MAC address is constructed for use in the SNPA field of TRILL Neighbor TLVs [RFC7176][RFC7177] on the link. This synthetic SNPA is as follows:



This synthetic SNPA/MAC address has the local (0x02) bit on in the first byte and so cannot conflict with any globally unique 48-bit Ethernet MAC. However, at the IP level, where TRILL operates on an IP link, there are only IP stations, not MAC stations, so conflict on the link with a real MAC address would be impossible in any case.

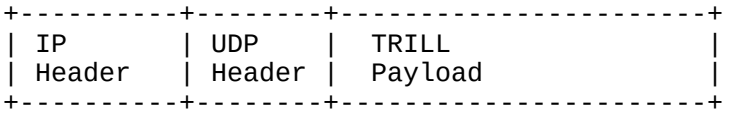
6. Port Configuration

Each RBridge physical port used for a TRILL over IP link MUST have at least one IP (v4 or v6) address. Implementations MAY allow a single physical port to operate as multiple IPv4 and/or IPv6 logical ports. Each IP address constitutes a different logical port and the RBridge with those ports MUST associate a different Port ID with each logical port.

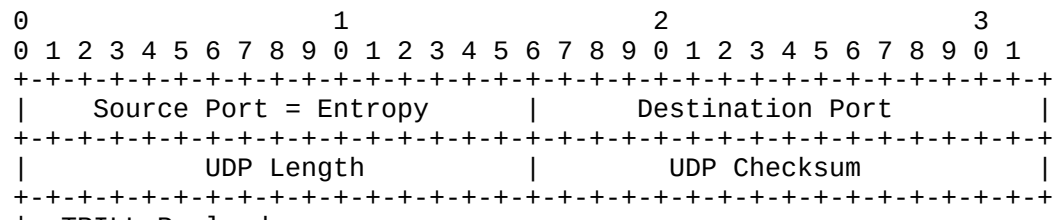
TBD: MUST be able to configure a list of IP addresses for serial unicast. MUST be able to configure a non-standard IP multi-cast address if native multicast is being used.

7. TRILL over UDP/IP Format

The general format of a TRILL over UDP/IP packet is shown below.



Where the UDP Header is as follows:



Source Port - see Section 10.2

Destination Port - indicates TRILL Data or IS-IS, see Section 14

UDP Length - as specified in [RFC768]

UDP Checksum - as specified in [RFC768]

The TRILL Payload starts with the TRILL Header (not including the TRILL Ethertype) for TRILL Data packets and starts with the 0x83 Intradomain Routeing Protocol Discriminator byte (thus not including the L2-IS-IS Ethertype) for TRILL IS-IS packets.

8. Handling Multicast

By default, both TRILL IS-IS packets and multi-destination TRILL Data packets are sent to an All-RBridges IPv4 or IPv6 multicast Address as appropriate (see Section 14); however, a TRILL over IP port may be configured to use serial unicast with a list of one or more unicast IP addresses of other TRILL over IP ports to which multi-destination packets are sent. Such configuration is necessary if the TRILL over IP port is connected to an IP network that does not support IP multicast. In both cases, unicast TRILL data packets would be sent by unicast IP.

When a TRILL over IP port is using IP multicast, it MUST periodically transmit appropriate IGMP (IPv4 [RFC3376]) or MLD (IPv6 [RFC2710]) packets so that the TRILL multicast IP traffic will be sent to it.

Although TRILL fully supports broadcast links with more than 2 RBridges connected to the, even where native IP multicast is available, there may be good reasons for configuring TRILL over IP ports to use unicast. In some networks, unicast is more reliable than multicast. If multiple unicast connections between parts of a TRILL campus are configured, TRILL will in any case spread traffic across them, treating them as parallel links, and appropriately fail

over traffic if a link ceases to operate or incorporate a new link that comes up.

9. Use of DTLS

All RBridges that support TRILL over IP MUST implement DTLS and support the use of DTLS to secure both TRILL IS-IS and TRILL data packets. When DTLS is used to secure a TRILL over IP link and no IS-IS security is enabled, the DTLS session MUST be fully established before any TRILL IS-IS or data packets are exchanged. When there is IS-IS security [RFC5304] or [RFC5310] provided, people may select to use IS-IS security to protect the IS-IS packet. Note that [RFC5304] only support MD5, which is not suggested to use at more. However, in this case, the DTLS session still MUST be fully established before any data packets transmission since IS-IS security does not provide any protection to data packets.

RBridges that implement TRILL over IP SHOULD support the use of certificates for DTLS and, if they support certificates, MUST support the following algorithm:

- TLS_RSA_WITH_AES_128_CBC_SHA256 [RFC5246]

RBridges that support TRILL over IP MUST support the use of pre-shared keys for DTLS. If the communicating RBridges have IS-IS Hello authentication enabled with a pre-shared key, then, by default a key derived from that TRILL Hello pre-shared key is used for DTLS unless some other pre-shared key is configured. The following cryptographic algorithms MUST be supported for use with pre-shared keys:

- TLS_PSK_WITH_AES_128_CBC_SHA256[RFC5487]

When applying pre-shared keys, a key needs to be derived from the default pre-shared key for DTLS usage. Specifically, the key is derived as follows:

HMAC-SHA256 ("TRILL IP"| IS-IS-shared key)

In the above "|" indicates concatenation, HMAC-SHA256 is as described in [FIPS180] [RFC6234] and "TRILL IP" is the eight byte US ASCII [ASCII] string indicated. When [RFC5310] is deployed, there could be multiple keys identified with 16-bit key IDs. In this case, the Key ID of IS-IS-shared key is also used to identify the derived key.

10. Transport Considerations

10.1. Recursive Ingress

TRILL is designed to transport end station traffic to and from IEEE 802.1Q conformant end stations and IP is frequently transported over IEEE 802.3 or similar protocols supporting 802.1Q conformant end stations. Thus, an end station data frame EF might get TRILL ingressed to TRILL(EF) which was then sent on a TRILL over IP over an 802.3 link resulting in an 802.3 frame of the form 802.3(IP(TRILL(EF))). There is a risk of such a packet being re-ingressed by the same TRILL campus, due to physical or logical misconfiguration, looping round, being further re-ingressed, etc. The packet might get discarded if it got too large but if fragmentation is enabled, it would just keep getting split into fragments that would continue to loop and grow and re-fragment until the path was saturated with junk and packets were being discarded due to queue overflow. The TRILL Header TTL would provide no protection because each TRILL ingress adds a new Header and TTL.

To protect against this scenario, TRILL over IP output ports MUST by, default, test whether a TRILL packet they are above to send is, in fact a TRILL ingress of a TRILL over IP over 802.3 or the like packets. That is, is it of the form TRILL(802.3(IP(TRILL(...)))? If so, the default action of the TRILL over IP output port is to discard the packet. However, there are cases where some level of nested ingress is desired so it MUST be possible to configure the port to allow such packets.

10.2. Fat Flows

For the purpose of load balancing, it is worthwhile to consider how to transport the TRILL packets over the Equal Cost Multiple Paths (ECMPs) existing in the IP path.

The ECMP election for the IP traffics could be based, at least for IPv4, on the quintuple of the outer IP header { Source IP, Destination IP, Source Port, Destination Port, and IP protocol }. Such tuples, however, can be exactly the same for all TRILL Data packets between two RBridge ports, even if there is a huge amount of data being sent. Therefore, in order to support ECMP, a RBridge SHOULD set the Source Port as an entropy field for ECMP decisions. This idea is also introduced in [I-D.yong-tsvwg-gre-in-udp-encap].

10.3. Congestion Considerations

TRILL can carry many different protocols as a payload. When a TRILL over IP flow carries primarily IP-based traffic, the aggregate traffic is assumed to be TCP friendly due to the congestion control mechanisms used by the payload traffic. Packet loss will trigger the necessary reduction in offered load, and no additional congestion avoidance action is necessary. When a TRILL over IP flow carries payload traffic that is not known to be TCP friendly and the flow runs across a path that could potentially become congested, additional mechanisms MUST be employed to ensure that the offered load on the TRILL link over IP is reduced appropriately during periods of congestion. This is not necessary in the case of a TRILL link over IP through an over- provisioned network, where the potential for congestion is avoided through the over-provisioning of the network.

11. MTU Considerations

In TRILL each RBridge advertises the largest LSP frame it can accept (but not less than 1,470 bytes) on any of its interfaces (at least those interfaces with adjacencies to other RBridges in the campus) in its LSP number zero through the originatingLSPBufferSize TLV [RFC6325] [RFC7176]. The campus minimum MTU, denoted Sz, is then established by taking the minimum of this advertised MTU for all RBridges in the campus. Links that do not meet the Sz MTU are not included in the routing topology. This protects the operation of IS-IS from links that would be unable to accommodate some LSPs.

A method of determining originatingLSPBufferSize for an RBridge with one or more TRILL over IP portsis described in [RFC7180]. However, if an IP link either can accommodate jumbo frames or is a link on which IP fragmentation is enabled and acceptable, then it is unlikely that the IP link will be a constraint on the RBridge’s originatingLSPBufferSize. On the other hand, if the IP link can only handle smaller frames and fragmentation is to be avoided when possible, a TRILL over IP port might constrain the RBridge’s originatingLSPBufferSize. Because TRILL sets the minimum values of Sz at 1,470 bytes, there may be links that meet the minimum MTU for the IP protocol (1,280 bytes for IPv6, theoretically 68 bytes for IPv4) on which it would be necessary to enable fragmentation for TRILL use.

The optional use of TRILL IS-IS MTU PDUs, as specified in [RFC6325] and [RFC7177] can provide added assurance of the actual MTU of a link.

12. Middlebox Considerations

TBD

13. Security Considerations

TRILL over IP is subject to all of the security considerations for the base TRILL protocol [RFC6325]. In addition, there are specific security requirements for different TRILL deployment scenarios, as discussed in the "Use Cases for TRILL over IP" section above.

This document specifies that all RBridges that support TRILL over IP MUST implement DTLS, and makes it clear that it is both wise and good to use DTLS in all cases where a TRILL over IP link will traverse a network that is not under the same administrative control as the rest of the TRILL campus. DTLS is necessary, in these cases to protect the privacy and integrity of data traffic.

TRILL over IP is completely compatible with the use of IS-IS security, which can be used to authenticate RBridges before allowing them to join a TRILL campus. This is sufficient to protect against rogue RBridges, but is not sufficient to protect data packets that may be sent, in UDP/IP tunnels, outside of the local network, or even across the public Internet. To protect the privacy and integrity of that traffic, use DTLS.

In cases were DTLS is used, the use of IS-IS security may not be necessary, but there is nothing about this specification that would prevent using both DTLS and IS-IS security together. In cases where both types of security are enabled, by default, a key derived from the IS-IS key will be used for DTLS.

14. IANA Considerations

IANA has allocated the following destination UDP Ports for the TRILL IS-IS and Data channels:

UDP Port	Protocol
(TBD)	TRILL IS-IS Channel
(TBD)	TRILL Data Channel

IANA has allocated one IPv4 and one IPv6 multicast address, as shown below, which correspond to the All-RBridges and All-IS-IS-RBridges multicast MAC addresses that the IEEE Registration Authority has assigned for TRILL. Because the low level hardware MAC address

dispatch considerations for TRILL over Ethernet do not apply to TRILL over IP, one IP multicast address for each version of IP is sufficient.

[Values recommended to IANA:]

Name	IPv4	IPv6
All-RBridges	233.252.14.0	FF0X:0:0:0:0:0:0:205

Note: when these IPv4 and IPv6 multicast addresses are used and the resulting IP frame is sent over Ethernet, the usual IP derived MAC address is used.

[Need to discuss scopes for IPv6 multicast (the "X" in the addresses) somewhere. Default to "site" scope but MUST be configurable?]

15. Acknowledgements

This document was written using the xml2rfc tool described in RFC 2629 [RFC2629].

The following people have provided useful feedback on the contents of this document: Sam Hartman, Adrian Farrel.

Some material has been derived from draft-ietf-mpls-in-udp by Xiaohu Xu, Nischal Sheth, Lucy Yong, Carlos Pignataro, and Yongbing Fan.

16. References

16.1. Normative References

[ASCII] "American National Standards Institute (formerly United States of America Standards Institute), "USA Code for Information Interchange", ANSI X3.4-1968, ANSI X3.4-1968 has been replaced by newer versions with slight modifications, but the 1968 version remains definitive for the Internet.", 1968.

[FIPS180] ""Secure Hash Standard (SHS)", United States of American, National Institute of Science and Technology, Federal Information Processing Standard (FIPS) 180-4", March 2012.

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

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, October 2008.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

[RFC5487] Badra, M., "Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode", RFC 5487, March 2009.

[RFC6325] Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

[RFC7176] Eastlake, D., Senevirathne, T., Ghanwani, A., Dutt, D., and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", RFC 7176, May 2014.

[RFC7177] Eastlake, D., Perlman, R., Ghanwani, A., Yang, H., and V. Manral, "Transparent Interconnection of Lots of Links (TRILL): Adjacency", RFC 7177, May 2014.

[RFC7180] Eastlake, D., Zhang, M., Ghanwani, A., Manral, V., and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL): Clarifications, Corrections, and Updates", RFC 7180, May 2014.

16.2. Informative References

[I-D.ietf-trill-fine-labeling]
Eastlake, D., Zhang, M., Agarwal, P., Perlman, R., and D. Dutt, "TRILL (Transparent Interconnection of Lots of Links): Fine-Grained Labeling", draft-ietf-trill-fine-labeling-07 (work in progress), May 2013.

[I-D.yong-tsvwg-gre-in-udp-encap]
Crabbe, E., Yong, L., and X. Xu, "Generic UDP Encapsulation for IP Tunneling", draft-yong-tsvwg-gre-in-udp-encap-02 (work in progress), October 2013.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC6234] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC6361] Carlson, J. and D. Eastlake, "PPP Transparent Interconnection of Lots of Links (TRILL) Protocol Control Protocol", RFC 6361, August 2011.

[RFC7172] Eastlake, D., Zhang, M., Agarwal, P., Perlman, R., and D. Dutt, "Transparent Interconnection of Lots of Links (TRILL): Fine-Grained Labeling", RFC 7172, May 2014.

[RFC7173] Yong, L., Eastlake, D., Aldrin, S., and J. Hudson, "Transparent Interconnection of Lots of Links (TRILL) Transport Using Pseudowires", RFC 7173, May 2014.

Authors' Addresses

Margaret Wasserman
Painless Security
356 Abbott Street
North Andover, MA 01845
USA

Phone: +1 781 405-7464
Email: mrw@painless-security.com
URI: http://www.painless-security.com

Donald Eastlake
Huawei Technologies
155 Beaver Street
Milford, MA 01757
USA

Phone: +1 508 333-2270
Email: d3e3e3@gmail.com

Dacheng Zhang
Huawei Technologies
Q14, Huawei Campus
No.156 Beiqing Rd.
Beijing, Hai-Dian District 100095
P.R. China

Email: zhangdacheng@huawei.com

TRILL Working Group
Internet-Draft
Intended Status: Standards Track
Expires: April 29, 2015

H. Zhai
JIT
T. Senevirathne
Cisco Systems
R. Perlman
EMC
M. Zhang
Y. Li
Huawei

October 26, 2014

TRILL: Pseudo-Nickname for Active-Active Access
draft-ietf-trill-pseudonode-nickname-02

Abstract

The IETF TRILL (TRansparent Interconnection of Lots of Links) protocol provides support for flow level multi-pathing for both unicast and multi-destination traffic in networks with arbitrary topology. Active-active access at the TRILL edge is the extension of these characteristics to end stations that are multiply connected to a TRILL campus as discussed in RFC 7379. In this document, the edge RBridge (TRILL switch) group providing active-active access to such an end station can be represented as a Virtual RBridge. Based on the concept of Virtual RBridge along with its pseudo-nickname, this document specifies a method for the TRILL active-active access by such end stations.

Status of This Memo

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Table of Contents

1. Introduction

The IETF TRILL protocol [RFC6325] provides optimal pair-wise data frame forwarding without configuration, safe forwarding even during periods of temporary loops, and support for multi-pathing of both unicast and multicast traffic. TRILL accomplishes this by using IS-IS [IS-IS] [RFC7176] link state routing and encapsulating traffic using a header that includes a hop count. Devices that implement TRILL are called RBridges or TRILL switches.

In the TRILL protocol, an end node can be attached to the TRILL campus via a point-to-point link or a shared link such as a bridged LAN (Local Area Network). Although there might be more than one edge RBridge on a shared link, to avoid potential forwarding loops, one and only one of the edge RBridges is permitted to provide forwarding service for end station traffic in each VLAN (Virtual LAN). That RBridge is referred to as the Appointed Forwarder (AF) for that VLAN on the link [RFC6325] [RFC6439]. However, in some practical deployments, to increase the access bandwidth and reliability, an end station might be multiply connected to several edge RBridges and all of the uplinks are handled via a Local Active-Active Link Protocol (LAALP) such as a Multi-Chassis Link Aggregation (MC-LAG) bundle [RFC7379]. In this case, it’s required that traffic can be ingressed/egressed into/from the TRILL campus by any of the RBridges for each given VLAN. These RBridges constitutes an Active-Active Edge (AAE) RBridge group.

With an LAALP, traffic with the same VLAN and source MAC address but belonging to different flows will frequently be sent to different member RBridges of the AAE group and then ingressed into TRILL campus. When an egress RBridge receives such TRILL data packets ingressed by different RBridges, it learns different VLAN and MAC address to nickname correspondences continuously when decapsulating the packets if it has data plane address learning enabled. This issue is known as the "MAC flip-flopping" issue, which makes most TRILL switches behave badly and causes the returning traffic to reach the destination via different paths resulting in persistent re-ordering of the frames. In addition to this issue, other issues such as duplicate egressing and loop back of multi-destination frames may also disturb the end stations multiply connected to the member RBridges of an AAE group [RFC7379].

Edge RBridge groups, which can be represented as a Virtual RBridge (RBv) and assigned a pseudo-nickname, address the AAE issues of TRILL and are specified in this document. A member RBridge of such a group uses a pseudo-nickname, instead of its own nickname, as the ingress RBridge nickname when ingressing frames received on attached LAALP links.

The main body of this document is organized as follows: Section 2 gives an overview of the TRILL active-active access issues and the reason that a virtual RBridge (RBv) is used to resolve the issues. Section 3 gives the concept of a virtual RBridge (RBv) and its pseudo-nickname. Section 4 describes how edge RBridges can support an RBv automatically and get a pseudo-nickname for the RBv. Section 5 discusses how to protect multi-destination traffic against disruption due to Reverse Forwarding Path (RPF) check failure, duplication, forwarding loop, etc. Section 6 covers the special processing of native frames and TRILL data packets at member RBridges of an RBv (also referred to as an Active-Active Edge (AAE) RBridge group). Section 7 describes the MAC information synchronization among the member RBridges of an RBv. Section 8 discusses protection against downlink failure at a member RBridge; and Section 9 gives the necessary IS-IS code points and data structures for a pseudo-nickname AAE RBridge group.

1.1. Terminology and Acronyms

This document uses the acronyms and terms defined in [RFC6325] and [RFC7379] and the following additional acronyms:

AAE - Active-active Edge RBridge group, a group of edge RBridges to which at least one CE is multiply attached with an LAALP. AAE is also referred to as edge group or Virtual RBridge in this document.

CE - Customer Equipment (end station or bridge). The device can be either physical or virtual equipment.

Data Label - VLAN FGL.

FGL - Fine-Grained Labeling or Fine-Grained Labeled or Fine-Grained Label [RFC7172].

LAALP - Local Active-Active Link Protocol [FRC7379].

OE flag - A flag used by the member RBridge of an LAALP to tell other edge RBridges whether it is willing to share an RBv with other LAALPs if they multiply attach to the same set of edge RBridges as it. When this flag for an LAALP is 1, it means that the LAALP needs to be served by an RBv by itself and is not willing to share, that is, it should Occupy an RBv Exclusively (OE).

RBv - virtual RBridge, an alias for active-active edge RBridge group in this document.

vDRB - The Designated RBridge in an RBv. It is responsible for

deciding the pseudo-nickname for the RBv.

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].

2. Overview

To minimize impact during failures and maximize available access bandwidth, Customer Equipment (referred to as CE in this document) may be multiply connected to TRILL campus via multiple edge RBridges. Figure 1 shows such a typical deployment scenario, where CE1 attaches to RB1, RB2, ... RBk and treats all of the uplinks as an LAALP bundle. Then RB1, RB2, ... RBk constitute an Active-active Edge (AAE) RBridge group for CE1 in this LAALP. Even if a member RBridge or an uplink fails, CE1 will still get frame forwarding service from the TRILL campus if there are still member RBridges and uplinks available in the AAE group. Furthermore, CE1 can make flow-based load balancing across the available member links of the LAALP bundle in the AAE group when it communicates with other CEs across the TRILL campus [RFC7379].

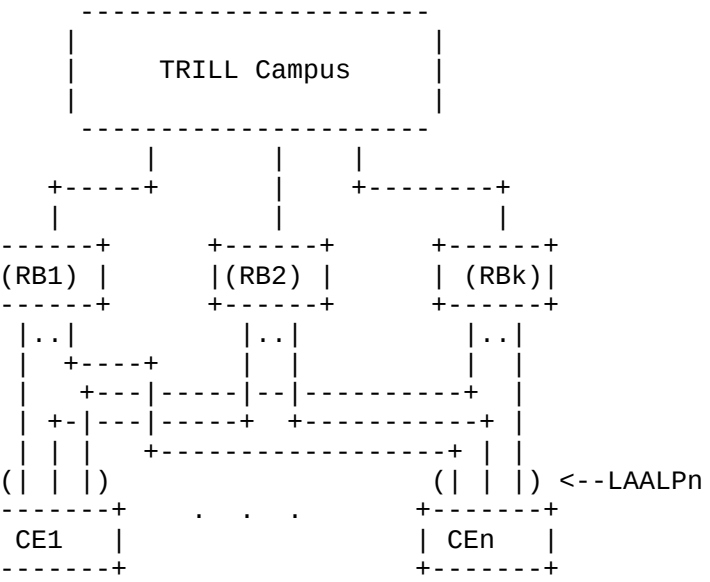


Figure 1 Active-Active Connection to TRILL Edge RBridges

By design, an LAALP (say LAALP1) does not forward packets received on one member port to other member ports. As a result, the TRILL Hello

messages sent by one member RBridge (say RB1) via a port to CE1 will not be forwarded to other member RBridges by CE1. That is to say, member RBridges will not see each other's Hellos via the LAALP. So every member RBridge of LAALP1 thinks of itself as appointed forwarder for all VLANs enabled on an LAALP1 link and can ingress/egress frames simultaneously in these VLANs.

The simultaneous flow-based ingressing/egressing can cause some problems. For example, simultaneous egressing of multi-destination traffic by multiple member RBridges will result in frame duplication at CE1 (see Section 3.1 of [RFC7379]); simultaneous ingressing of frames originated by CE1 for different flows in the same VLAN with the same source MAC address will result in MAC address flip-flopping at remote egress RBridges that have data plane address learning enabled (see Section 3.3 of [RFC7379]). The flip-flopping would in turn cause packet re-ordering in reverse traffic.

Edge RBridges learn Data Label and MAC address to nickname correspondences by default via decapsulating TRILL data packets (see Section 4.8.1 of [RFC6325] as updated by [RFC7172]). The MAC flip-flopping issue is solved herein based on the assumption that the default learning is enabled at edge RBridges, so this document specifies Virtual RBridge, together with its pseudo-nickname.

3. Virtual RBridge and its Pseudo-nickname

A Virtual RBridge (RBv) represents a group of edge RBridges to which at least one CE is multiply attached using LAALP. More exactly, it represents a group of ports on the edge RBridges providing end station service and the service provided to the CE(s) on these ports, through which the CE(s) are multiply attached to TRILL campus using LAALP(s). Such end station service ports are called RBv ports; in contrast, other access ports at edge RBridges are called regular access ports in this document. RBv ports are always LAALP connecting ports, but not vice versa (see Section 4.1). For an edge RBridge, if one or more of its end station service ports are ports of an RBv, that RBridge is a member RBridge of that RBv.

For the convenience of description, a Virtual RBridge is also referred to as an Active-Active Edge (AAE) group in this document. In the TRILL campus, an RBv is identified by its pseudo-nickname, which is different from any RBridge's regular nickname(s). An RBv has one and only one pseudo-nickname. Each member RBridge (say RB1, RB2 ..., RBk) of an RBv (say RBvn) advertises RBvn's pseudo-nickname using a Nickname sub-TLV in its TRILL IS-IS LSP (Link State PDU) [RFC7176] and SHOULD do so with maximum priority of use (0xFF), along with their regular nickname(s). (Maximum priority is recommended to avoid

the disruption to AAE group that would occur if the nickname were taken away by a higher priority RBridge.) Then, from these LSPs, other RBridges outside the AAE group know that RBvn is reachable through RB1 to RBk.

A member RBridge (say RBi) loses its membership from RBvn when its last port in RBvn becomes unavailable due to failure, re-configuration, etc. Then RBi removes RBvn's pseudo-nickname from its LSP and distributes the updated LSP as usual. From those updated LSPs, other RBridges know that there is no path to RBvn through RBi now.

When member RBridges receive native frames from their RBv ports and decide to ingress the frames into the TRILL campus, they use that RBv's pseudo-nickname instead of their own regular nicknames as the ingress nickname to encapsulate them into TRILL Data packets. So when these packets arrive at an egress RBridge, even if they are originated by the same end station in the same VLAN but ingressed by different member RBridges, no address flip-flopping is observed on the egress RBridge when decapsulating these packets. (When a member RBridge of an AAE group ingresses a frame from a non-RBv port, it still uses its own regular nickname as the ingress nickname.)

Since RBv is not a physical node and no TRILL frames are forwarded between its ports via a LAALP, pseudo-node LSP(s) MUST NOT be created for an RBv. RBv cannot act as a root when constructing distribution trees for multi-destination traffic and its pseudo-nickname is ignored when determining the distribution tree root for TRILL campus [CMT]. So the tree root priority of RBv's nickname MUST be set to 0, and this nickname SHOULD NOT be listed in the "s" nicknames (see Section 2.5 of [RFC6325]) by the RBridge holding the highest priority tree root nickname.

NOTE: In order to reduce the consumption of nicknames, especially in large TRILL campus with lots of RBridges and/or active-active accesses, when multiple CEs attach to the exact same set of edge RBridges via LAALPs, those edge RBridges should be considered as a single RBv with a pseudo-nickname.

4. Member RBridges Auto-Discovery

Edge RBridges connected to a CE via an LAALP can automatically discover each other with minimal configuration through exchange of LAALP connection information.

From the perspective of edge RBridges, a CE that connects to edge RBridges via an LAALP can be identified by the ID of the LAALP that

is unique across the TRILL campus (for example, the MC-LAG System ID [802.1AX]), which is referred to as LAALP ID in this document. On each of such edge RBridges, the access port to such a CE is associated with an LAALP ID for the CE. An LAALP is considered valid on an edge RBridge only if the RBridge still has operational down-link to that LAALP. For such an edge RBridge, it advertises a list of LAALP IDs for its valid local LAALPs to other edge RBridges via its E-L1FS FS-LSP(s) [rfc7180bis]. Based on the LAALP IDs advertised by other RBridges, each RBridge can know which edge RBridges could constitute an AAE group (See Section 4.1 for more details). Then one RBridge is elected from the group to allocate an available nickname (the pseudo-nickname) for the group (See Section 4.2 for more details).

4.1. Discovering Member RBridge for an RBv

Take Figure 2 as an example, where CE1 and CE2 multiply attach to RB1, RB2 and RB3 via LAALP1 and LAALP2 respectively; CE3 and CE4 attach to RB3 and RB4 via LAALP3 and LAALP4 respectively. Assume LAALP3 is configured to occupy a Virtual RBridge by itself.

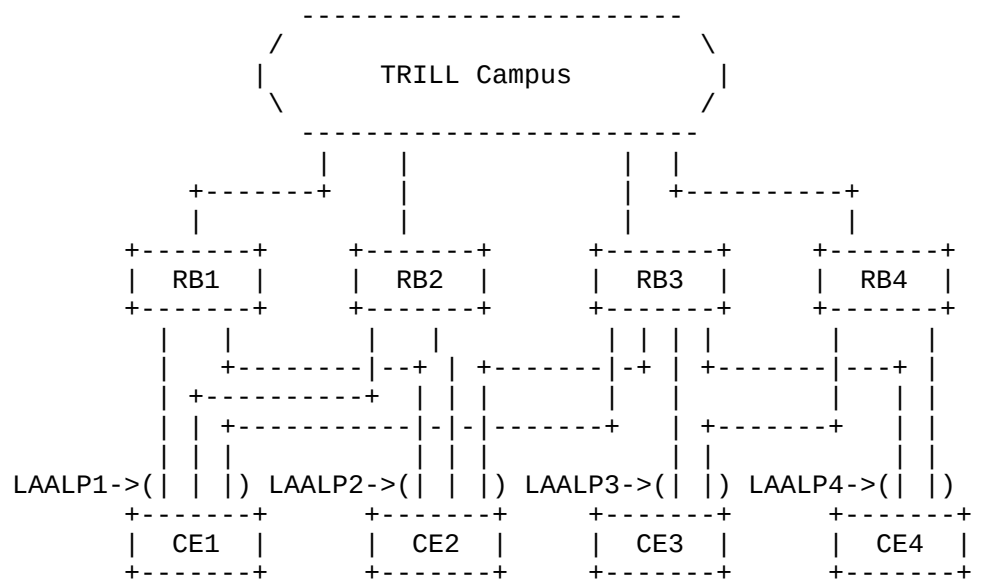


Figure 2 Different LAALPs to TRILL Campus

RB1 and RB2 advertise {LAALP1, LAALP2} in the LAALP Membership sub-TLV (see Section 9.1 for more details) via their TRILL IS-IS LSPs respectively; RB3 announces {LAALP1, LAALP2, LAALP3, LAALP}; and RB4 announces {LAALP3, LAALP4}, respectively.

An edge RBridge is called an LAALP related RBridge if it has at least one LAALP configured on an access port. On receipt of the LAALP Membership sub-TLVs, RBn ignores them if it is not an LAALP related RBridge; otherwise, RBn SHOULD use the LAALP information contained in the sub-TLVs, along with its own LAALP Membership sub-TLVs to decide which RBv(s) it should join and which edge RBridges constitute each of such RBvs. Based on the information received, each of the 4 RBridges knows the following information:

LAALP ID	OE-flag	Set of edge RBridges
LAALP1	0	{RB1, RB2, RB3}
LAALP2	0	{RB1, RB2, RB3}
LAALP3	1	{RB3, RB4}
LAALP4	0	{RB3, RB4}

Where the OE-flag indicates whether an LAALP is willing to share an RBv with other LAALPs if they multiply attach to exact the same set of edge RBridges as it. For an LAALP (for example LAALP3), if its OE-flag is one, it means that LAALP3 does not want to share, so it MUST Occupy an RBv Exclusively (OE).

Otherwise, the LAALP (for example LAALP1) will share an RBv with other LAALPs if possible. By default, this flag is set zero. For an LAALP, this flag is considered 1 only if any edge RBridge advertises it as one (see Section 9.1).

In the above table, there might be some LAALPs that attach to a single RBridge due to mis-configuration or link failure, etc. Those LAALPs are considered as invalid entries. Then each of the LAALP related edge RBridges performs the following algorithm to decide which valid LAALPs can be served by an RBv.

Step 1: Take all the valid LAALPs that have their OE-flags set to 1 out of the table and create an RBv per such LAALP.

Step 2: Sort the valid LAALPs left in the table in descending order based on the number of RBridges in their associated set of multi-homed RBridges. In the case that several LAALPs have same number of RBridges, these LAALPs are then ordered in ascending order in the proper places of the table based on their LAALP IDs considered as unsigned integers. (for example, in the above table, both LAALP1 and LAALP2 have 3 member RBridges, assuming LAALP1 ID is smaller than LAALP2 ID, so LAALP1 is followed by LAALP2 in the ordered table.)

Step 3: Take the first valid LAALP (say LAALP_i) with the maximum set of RBridges, say S_i, out of the table and create a new RBv (Say RBv_i) for it.

Step 4: Walk through the remaining valid LAALPs in the table one by one, pick up all the valid LAALPs that their sets of multi-homed RBridges contain exactly the same RBridges as that of LAALP_i and take them out of the table. Then appoint RBv_i as the servicing RBv for those LAALPs.

Step 5: Repeat Step 3-4 for the left LAALPs until all the valid entries in the table has be associated with an RBv.

After performing the above steps, all the 4 RBridges know that LAALP3 is served by an RBv, say RBv1, which has RB3 and RB4 as member RBrdges; LAALP1 and LAALP2 are served by another RBv, say RBv2, which has RB1, RB2 and RB3 as member RBridges; and LAALP4 is served by RBv3, which has RB3 and RB4 as member RBridges, shown as follows:

RBv	Serving LAALPs	Member RBridges
RBv1	{LAALP3}	{RB3, RB4}
RBv2	{LAALP1, LAALP2}	{RB1, RB2, RB3}
RBv3	{LAALP4}	{RB3, RB4}

In each RBv, one of the member RBridges is elected as the DRB (Designated RBridge) of the RBv. Then this RBridge picks up an available nickname as the pseudo-nickname for the RBv and announces it to all other member RBridges of the RBv via its TRILL IS-IS LSPs (refer to Section 9.2 for the relative extended sub-TLVs).

4.2. Selection of Pseudo-nickname for RBv

As described in Section 3, in the TRILL campus, an RBv is identified by its pseudo-nickname. In an AAE group (i.e., RBv), one member RBridge is elected for the duty to select a pseudo-nickname for this RBv; this RBridge is called Designated RBridge of the RBv (vDRB) in this document. The winner is the RBridge with the largest IS-IS System ID considered as an unsigned integer, in the group. Then based on its TRILL IS-IS link state database and the potential pseudo-nickname(s) reported in the LAALP Membership sub-TLVs by other member RBridges of this RBv (see Section 9.1 for more details), the vDRB selects an available nickname as the pseudo-nickname for this RBv and advertizes it to the other RBridges via its E-L1FS FS-LSP(s) (see Section 9.2 and [rfc7180bis]). Except as provided below, the selection of a nickname to use as the pseudo-nickname follows the usual TRILL rules given in [RFC6325] as updated by [rfc7180bis]. On receipt of the pseudo-nickname advertised by the vDRB, all the other RBridges of that group associate it with the LAALPs served by the RBv, and then download the association to their data plane fast path logic.

To reduce the traffic disruption caused by nickname changing, if possible, vDRB SHOULD attempt to reuse the pseudo-nickname recently used by the group when selecting nickname for the RBv. To help the vDRB to do so, each LAALP related RBridge advertises a re-using pseudo-nickname for each of its LAALPs in its LAALP Membership sub-TLV if it has used such a pseudo-nickname for that LAALP recently. Although it is up to the implementation of the vDRB as to how to treat the re-using pseudo-nicknames, the following is suggested:

- If there are multiple available re-using pseudo-nicknames that are reported by all the member RBridges of some LAALPs in this RBv, the available one that is reported by most of such LAALPs is chosen as the pseudo-nickname for this RBv. If a tie exists, the re-using pseudo-nickname with the smallest value considered as an unsigned integer is chosen.

- If only one re-using pseudo-nickname is reported, it SHOULD be chosen if available.

If there is no available re-using pseudo-nickname reported, the vDRB selects a nickname by its usual method.

Then the selected pseudo-nickname is announced by the vDRB to other member RBridges of this RBv in the PN-RBv sub-TLV (see Section 9.2). After receiving the pseudo-nickname, other RBridges of that RBv associate the nickname with their ports of that RBv and download the association to their data plane fast path logic.

5. Distribution Trees and Designated Forwarder

In an AAE group (i.e., an RBv), as each of the member RBridges thinks it is the appointed forwarder for VLAN x, without changes made for active-active connection support, they would all ingress/egress frames into/from TRILL campus for all VLANs. For multi-destination frames, more than one member RBridges ingressing them may cause some of the resulting TRILL Data packets to be discarded due to failure of Reverse Path Forwarding (RPF) Check on other RBridges; for a multi-destination traffic, more than one RBridges egressing it may cause local CE(s) receiving duplication frame. Furthermore, in an AAE group, a multi-destination frame sent by a CE (say CEi) may be ingressed into TRILL campus by one member RBridge, then another member RBridge will receive it from TRILL campus and egress it to CEi, which will result in loop back of frame for CEi. These problems are all described in [RFC7379].

In the following sub-sections, the first two issues are discussed in Section 5.1 and Section 5.2, respectively; the third one is discussed

in Section 5.3.

5.1. Different Trees for Different Member RBridges

In TRILL, RBridges use distribution trees to forward multi-destination frames (although under some circumstances they can be unicast as specified in [RFC7172]). An RPF Check along with other checking is used to avoid temporary multicast loops during topology changes (Section 4.5.2 of [RFC6325]). The RPF check mechanism only accepts a multi-destination frame ingressed by an RBridge RBi and forwarded on a distribution tree Tx if it arrives at another RBridge RBn on the expected port. If arriving on any other port, the frame MUST be dropped.

To avoid address flip-flopping on remote RBridges, member RBridges use RBv's pseudo-nickname instead of their regular nicknames as ingress nickname to ingress native frames, including multicast frames. From the view of other RBridges, these frames appear as if they were ingressed by the RBv. When multicast frames of different flows are ingressed by different member RBridges of an RBv and forwarded along the same distribution tree, they may arrive at RBn on different ports. Some of them will violate the RFC check principle at RBn and be dropped, which may result in traffic disruption.

In an RBv, if different member RBridge uses different distribution trees to ingress multi-destination frames, the RFC check violation issue can be fixed. Coordinated Multicast Trees (CMT) proposes such an approach, and makes use of the Affinity sub-TLV defined in [RFC7176] to tell other RBridges which trees a member RBridge (say RBi) may choose when ingressing multi-destination frames;then all RBridges in the TRILL campus can calculate RFC check information for RBi on those trees taking the tree affinity information into account [CMT].

This document specifies that the approach proposed in [CMT] will be used to fix the RFC check violation issue. Please refer to [CMT] for more details of the approach.

5.2. Designated Forwarder for Member RBridges

Take Figure 3 as an example, where CE1 and CE2 are served by an RBv that has RB1 and RB2 as member RBridges. In VLAN x, the three CEs can communicate with each other.

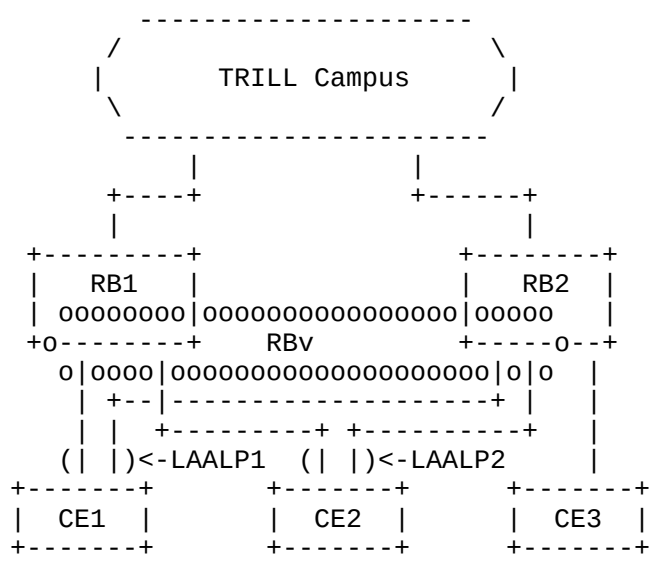


Figure 3 A Topology with Multi-homed and Single-homed CEs

When a remote RBridge (say RBn) sends a multi-destination TRILL Data packet in VLAN x (or the FGL that VLAN x maps to if the packet is an FGL one), both RB1 and RB2 will receive it. As each of them thinks it is the appointed forwarder for VLAN x, without changes made for active-active connection support, they would both forward the frame to CE1/CE2. As a result, CE1/CE2 would receive duplicate copies of the frame through this RBv.

In another case, assume CE3 is single-homed to RB2. When it transmits a native multi-destination frame onto link CE3-RB2 in VLAN x, the frame can be locally replicated to the ports to CE1/CE2, and also encapsulated into TRILL Data packet and ingressed into TRILL campus. When the packet arrives at RB1 across the TRILL campus, it will be egressed to CE1/CE2 by RB1. Then CE1/CE2 receives duplicate copies from RB1 and RB2.

In this document, Designated Forwarder (DF) for a VLAN is introduced to avoid the duplicate copies. The basic idea of DF is to elect one RBridge per VLAN from an RBv to egress multi-destination TRILL Data traffic and replicate locally-received multi-destination native frames to the CEs served by the RBv.

Note that DF has an effect only on the egressing/replicating of multi-destination traffic, no effect on the ingressing of frames or forwarding/egressing of unicast frames. Furthermore, DF check is performed only for RBv ports, not on regular access ports.

Each RBridge in an RBv elects a DF using same algorithm which guarantees the same RBridge elected as DF per VLAN.

Assuming there are m LAALPs and k member RBridges in an RBv; each LAALP is referred to as LAALPi where 0 <= i < m, and each RBridge is referred to as RBj where 0 <= j < k-1, DF election algorithm per VLAN is as follows:

Step 1: For LAALPi, sort all the RBridges in numerically ascending order based on (System IDj | LAALPi) mod k, where "System IDj" is the IS-IS System ID of RBj, "|" means concatenation, and LAALPi is the LAALP ID for LAALPi. In the case that some RBridges get the same result of the mod, these RBridges are sorted in numerically ascending order in the proper places of the result in the list by their System IDs.

Step 2: Each RBridge in the numerically sorted list is assigned a monotonically increasing number j, such that increasing number j corresponding to its position in the sorted list, i.e., the first RBridge (the first one with the smallest (System ID | LAALP ID) mod k) is assigned zero and the last is assigned k-1.

Step 3: For each VLAN ID n, choose the RBridge whose number equals (n mod k) as DF.

Step 4: Repeat Step 1-3 for the remaining LAALPs until there is a DF per VLAN per LAALP in the RBv.

For a multi-destination native frame of VLAN x received, if RBi is an LAALP attached RBridge, in addition to local replication of the frame to regular access ports as per [RFC6325] (and [RFC7172] for FGL), it MUST also locally replicate the frame to the following RBv ports when one of the following conditions is met:

1) RBv ports associated with the same pseudo-nickname as that of the incoming port, no matter whether RBi is the DF for the frame’s VLAN on the outgoing ports except that the frame MUST NOT be replicated back to the incoming port;

2) RBv ports on which RBi is the DF for the frame’s VLAN while they are associated with different pseudo-nickname(s) to that of the incoming port.

For non-LAALP related RBridges or for non-RBv ports on an LAALP related RBridge, local replication is performed as per [RFC6325].

For a multi-destination TRILL Data packet received, RBi MUST NOT egress it out of the RBv ports where it is not DF for the frame’s

Inner.VLAN (or for the VLAN corresponding to the Inner.Label if the packet is an FGL one). Otherwise, whether or not egressing it out of such ports is further subject to the filtering check result of the frame's ingress nickname on these ports (see Section 5.3).

5.3. Ingress Nickname Filtering

As shown in Figure 3, CE1 may send a multicast traffic in VLAN x to TRILL campus via a member RBridge (say RB1). The traffic is then TRILL-encapsulated by RB1 and delivered through TRILL campus to multi-destination receivers. RB2 may receive the traffic, and egress it back to CE1 if it is the DF for VLAN x on the port to LAALP1. Then the traffic loops back to CE1 (see Section 3.2 of [RFC7379).

To fix the above issue, an ingress nickname filtering check is required by this document. The idea of this check is to check the ingress nickname of a multi-destination TRILL Data packet before egressing a copy of it out of an RBv port. If the ingress nickname matches the pseudo-nickname of the RBv (associated with the port), the filtering check should fail and the copy MUST NOT be egressed out of that RBv port. Otherwise, the copy is egressed out of that port if it has also passed other checks, such as the appointed forwarder check in Section 4.6.2.5 of [RFC6325] and the DF check in Section 5.2.

Note that this ingress nickname filtering check has no effect on the multi-destination native frames received on access ports and replicated to other local ports (including RBv ports), since there is no ingress nickname associated with such frames. Furthermore, for the RBridge regular access ports, there is no pseudo-nickname associated with them; so no ingress nickname filtering check is required on those ports.

More details of data packet processing on RBv ports are given in the next section.

6. TRILL Traffic Processing

This section provides more details of native frame and TRILL Data packet processing as it relates to the RBv's pseudo-nickname.

6.1. Native Frames Ingressing

When RB1 receives a unicast native frame from one of its ports that has end-station service enabled, it processes the frame as described in Section 4.6.1.1 of [RFC6325] with the following exception.

- o If the port is an RBv port, RB1 uses the RBv's pseudo-nickname, instead of one of its regular nickname(s) as the ingress nickname when doing TRILL encapsulation on the frame.

When RB1 receives a native multi-destination (Broadcast, Unknown unicast or Multicast) frame from one of its access ports (including regular access ports and RBv ports), it processes the frame as described in Section 4.6.1.2 of [RFC6325] with the following exceptions.

- o If the incoming port is an RBv port, RB1 uses the RBv's pseudo-nickname, instead of one of its regular nickname(s) as the ingress nickname when doing TRILL encapsulation on the frame.

- o For the copies of the frame replicated locally to RBv ports, there are two cases as follows:

  - If the outgoing port(s) is associated with the same pseudo-nickname as that of the incoming port but not with the same LAALP as the incoming port, the copies are forwarded out of that outgoing port(s) after passing the appointed forwarder check for the frame's VLAN. That is to say, the copies are processed on such port(s) as Section 4.6.1.2 of [RFC6325].

  - Else, the Designated Forwarder (DF) check is further made on the outgoing ports for the frame's VLAN after the appointed forwarder check. The copies are not output through the ports that failed the DF check (i.e., RB1 is not DF for the frame's VLAN on the ports); otherwise, the copies are forwarded out of the ports that pass the DF check (see Section 5.2).

For such a frame received, the MAC address information learned by observing it, together with the LAALP ID of the incoming port SHOULD be shared with other member RBridges in the group (see Section 7).

6.2. Egressing TRILL Data Packets

This section describes egress processing of the TRILL Data packets received on an RBv member RBridge (say RBn). Section 6.2.1 describes the egress processing of unicast TRILL Data packets and Section 6.2.2 specifies the multi-destination TRILL Data packets egressing.

6.2.1. Unicast TRILL Data Packets

When receiving a unicast TRILL data packet, RBn checks the egress nickname in the TRILL header of the packet. If the egress nickname is one of RBn's regular nicknames, the packet is processed as defined in Section 4.6.2.4 of [RFC6325].

If the egress nickname is the pseudo-nickname of a local RBv, RBn is responsible for learning the source MAC address, unless data plane learning has been disabled. The learned {Inner.MacSA, Data Label, ingress nickname} triplet SHOULD be shared within the AAE group (See Section 7).

Then the packet is de-capsulated to its native form. The Inner.MacDA and Data Label are looked up in RBn's local forwarding tables, and one of the three following cases may occur. RBn uses the first case that applies and ignores the remaining cases:

- If the destination end station identified by the Inner.MacDA and Data Label is on a local link, the native frame is sent onto that link with the VLAN from the Inner.VLAN or VLAN corresponding to the Inner.Label if the packet is FGL.

- Else if RBn can reach the destination through another member RBridge RBk, it tunnels the native frame to RBk by re-encapsulating it into a unicast TRILL Data packet and sends it to RBk. RBn uses RBk's regular nickname, instead of the pseudo-nickname as the egress nickname for the re-encapsulation, and the ingress nickname remains unchanged (somewhat similar to Section 2.4.2.1 of [rfc7180bis]). If the hop count value of the packet is too small for it to reach RBk safely, RBn SHOULD increase that value properly in doing the re-encapsulation. (NOTE: When receiving that re-encapsulated TRILL Data packet, as the egress nickname of the packet is RBk's regular nickname rather than the pseudo-nickname of a local RBv, RBk will process it as Section 4.6.2.4 of [RFC6325], and will not re-forward it to another RBridge.)

- Else, RBn does not know how to reach the destination; it sends the native frame out of all the local ports on which it is appointed forwarder for the Inner.VLAN (or appointed forwarder for the VLAN into which the Inner.Label maps on that port for FGL TRILL Data packet [RFC7172]).

6.2.2. Multi-Destination TRILL Data Packets

When RB1 receives a multi-destination TRILL Data Packet, it checks and processes the packet as described in Section 4.6.2.5 of [RFC6325] with the following exception.

- On each RBv port where RBn is the appointed forwarder for the packet's Inner.VLAN (or for the VLAN to which the packet's Inner.Label maps on that port if it is an FGL TRILL Data packet), the Designated Forwarder check (see Section 5.2) and the Ingress Nickname Filtering check (see Section 5.3) are further performed.

For such an RBv port, if either the DF check or the filtering check fails, the frame MUST NOT be egressed out of that port. Otherwise, it can be egressed out of that port.

7. MAC Information Synchronization in Edge Group

An edge RBridge, say RB1 in LAALP1, may have learned a MAC address and Data Label to nickname correspondence for a remote host h1 when h1 sends a packet to CE1. The returning traffic from CE1 may go to any other member RBridge of LAALP1, for example RB2. RB2 may not have that correspondence stored. Therefore it has to do the flooding for unknown unicast. Such flooding is unnecessary since the returning traffic is almost always expected and RB1 had learned the address correspondence. To avoid the unnecessary flooding, RB1 SHOULD share the correspondence with other RBridges of LAALP1. RB1 synchronizes the correspondence by using the MAC-RI sub-TLV [RFC6165] in its ESADI-LSPs [RFC7357].

On the other hand, RB2 has learned the MAC and Data Label of CE1 when CE1 sends a frame to h1 through RB2. The returning traffic from h1 may go to RB1. RB1 may not have CE1’s MAC and Data Label stored even though it is in the same LAALP for CE1 as RB2. Therefore it has to flood the traffic out of all its access ports where it is appointed forwarder for the VLAN (see Section 6.2.1) or the VLAN the FGL maps to on that port if the packet is FGL. Such flooding is unnecessary since the returning traffic is almost always expected and RB2 had learned the CE1’s MAC and Data Label information. To avoid that unnecessary flooding, RB2 SHOULD share the MAC and VLAN (or MAC and FGL if the egress port is an FGL port [RFC7172]) with other RBridges of LAALP1. RB2 synchronizes the MAC and Data Label by enclosing the relative MAC-RI TLV with a pair of boundary TRILL APPsub-TLVs for LAALP1 (see Section 9.3) in its ESADI-LSP [RFC7357]. After receiving the enclosed MAC-RI TLVs, the member RBridges of LAALP1 (i.e., LAALP1 related RBridges) treat the MAC and Data Label as if it was learned by them locally on their member port of LAALP1; the LAALP1 unrelated RBridges just ignore LAALP1’s information contained in the boundary APPsub-TLVs and treat the MAC and Data Label as specified in [RFC7357]. Furthermore, in order to make the LAALP1 unrelated RBridges know that the MAC and Data Label is reachable through the RBv that provides service to LAALP1, the Topology-id/Nickname field of the MAC-RI TLV SHOULD carry the pseudo-nickname of the RBv rather than zero or one of the originating RBridge’s (i.e., RB2’s) regular nicknames.

8. Member Link Failure in RBv

As shown in Figure 4, suppose the link RB1-CE1 fails. Although a new RBv will be formed by RB2 and RB3 to provide active-active service for LAALP1 (see Section 5), the unicast traffic to CE1 might be still forwarded to RB1 before the remote RBridge learns CE1 is attached to the new RBv. That traffic might be disrupted by the link failure. Section 8.1 discusses the failure protection in this scenario.

However, for multi-destination TRILL Data packets, since they can reach all member RBridges of the new RBv and be egressed to CE1 by either RB2 or RB3 (i.e., the new DF for the traffic's Inner.VLAN or the VLAN the packet's Inner.Label maps to in the new RBv), special actions to protect against down-link failure for such multi-desination packets is not needed.

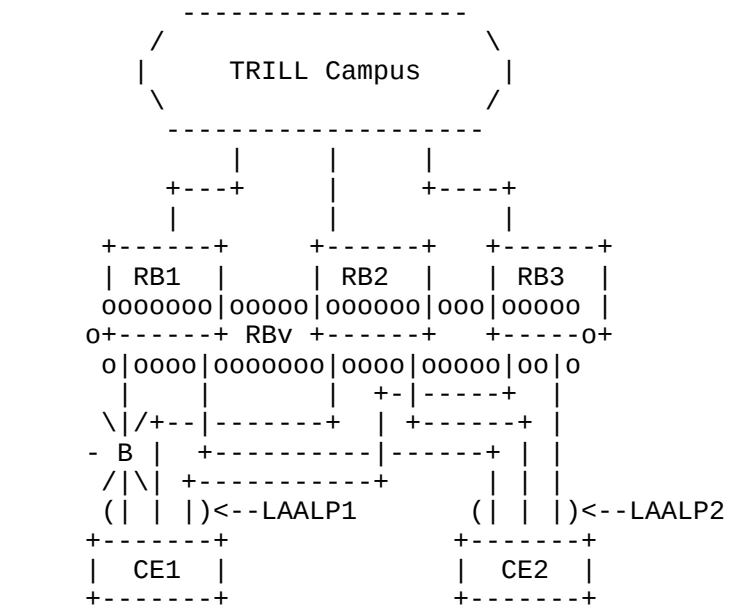


Figure 4 A Topology with Multi-homed and Single-homed CEs

8.1. Link Protection for Unicast Frame Egressing

When the link CE1-RB1 fails, RB1 loses its direct connection to CE1. The MAC entry through the failed link to CE1 is removed from RB1's local forwarding table immediately. Another MAC entry learned from another member RBridge of LAALP1 (for example RB2, since it is still a member RBridge of LAALP1) is installed into RB1's forwarding table (see Section 9.3). In that new entry, RB2 (identified by one of its regular nicknames) is the egress RBridge for CE1's MAC address. Then

when a TRILL Data packet to CE1 is delivered to RB1, it can be tunneled to RB2 after being re-encapsulated (ingress nickname remains unchanged and egress nickname is replaced by RB2's regular nickname) based on the above installed MAC entry (see bullet 2 in Section 6.2.1). Then RB2 receives the frame and egresses it to CE1.

After the failure recovery, RB1 learns that it can reach CE1 via link CE1-RB1 again by observing CE1's native frames or from the MAC information synchronization by member RBridge(s) of LAALP1 described in Section 7, then it restores the MAC entry to its previous one and downloads it to its data plane fast path logic.

9. TLV Extensions for Edge RBridge Group

9.1. LAALP Membership APPsub-TLV

This APPsub-TLV is used by an edge RBridge to announce its associated LAALP information. It is defined as a sub-TLV of the TRILL GENINFO TLV [RFC7357] and is distributed in E-L1FS FS-LSPs [rfc7180bis]. It has the following format:

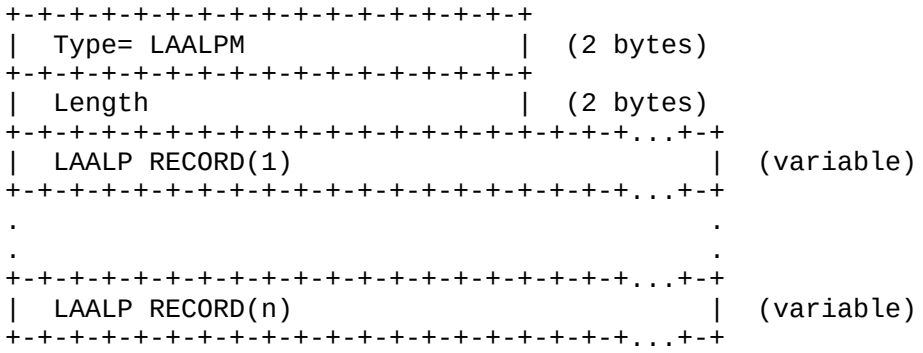
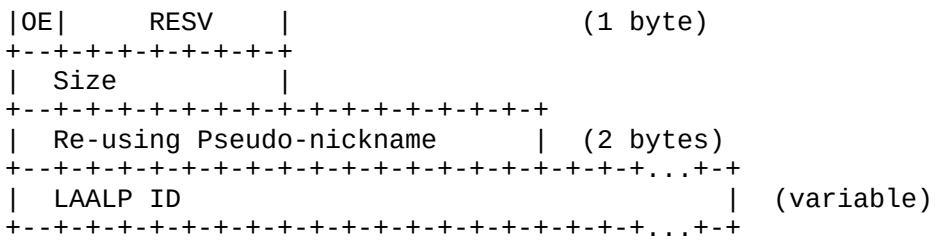


Figure 5 LAALP Membership Advertisement APPsub-TLV

where each LAALP RECORD has the following form:



- LAALPM (2 bytes): Defines the type of this sub-TLV, #tbd1.

- Length (2 bytes): the sum of the lengths of the LAALP RECORDs.

- OE (1 bit): an flag indicating whether or not the LAALP wants to occupy an RBv by itself; 1 for occupying by itself (or Occupying Exclusively (OE)). By default, it is set to 0 on transmit. This bit is used for edge RBridge group auto-discovery (see Section 4.1). For any one LAALP, the values of this flag might conflict in the LSPs advertised by different member RBridges of that LAALP. In that case, the flag for that LAALP is considered as 1.

- RESV (7 bits): MUST be transmitted as zero and ignored on receipt.

- Size (1 byte): Size of remaining part of LAALP RECORD (2 plus length of the LAALP ID).

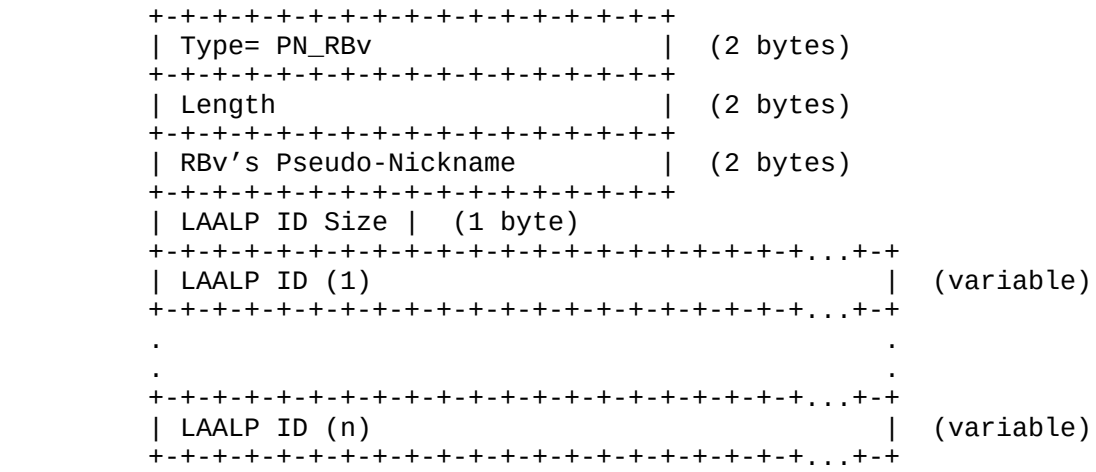
- Re-using Pseudo-nickname (2 bytes): Suggested pseudo-nickname of the AAE group serving the LAALP. If the LAALP is not served by any AAE group, this field MUST be set to zero. It is used by the originating RBridge to help the vDRB to reuse the previous pseudo-nickname of an AAE group (see Section 4.2).

- LAALP ID (variable): The ID of the LAALP. If the LAALP is an MC-LAG, it is the 8 byte ID as specified in Section 5.3.2 in [802.1AX].

On receipt of such an APPsub-TLV, if RBn is not an LAALP related edge RBridge, it ignores the sub-TLV; otherwise, it parses the sub-TLV. When new LAALPs are found or old ones are withdrawn compared to its old copy, and they are also configured on RBn, it triggers RBn to perform the "Member RBridges Auto-Discovery" procedure described in Section 4.1.

9.2. PN-RBV APPsub-TLV

The PN-RBv APPsub-TLV is used by a Designated RBridge of a Virtual RBridge (vDRB) to dictate the Pseudo-nickname for the LAALPs served by the RBv. It is defined as a sub-TLV of TRILL GENINFO TLV [RFC7357] and is distributed in E-L1FS FS-LSP [rfc7180bis]. It has the following format:



- PN_RBv (2 bytes): Defines the type of this sub-TLV, #tbd2.

- Length (2 bytes): 3+n*k bytes, where there are n LAALP IDs, each of size k bytes. k is found in the LLALP ID Size field below. If Length is not 3 plus an integer time k, the sub-TLV is corrupt and MUST be ignored.

- RBv's Pseudo-Nickname (2 bytes): The appointed pseudo-nickname for the RBv that serves for the LAALPs listed in the following fields.

- LAALP ID Size (1 byte): The size of each of the following LAALP IDs in this sub-TLV. 8 if the LAALPs listed are MC-LAGs. The value in this field is the k that appears in the formula for Length above.

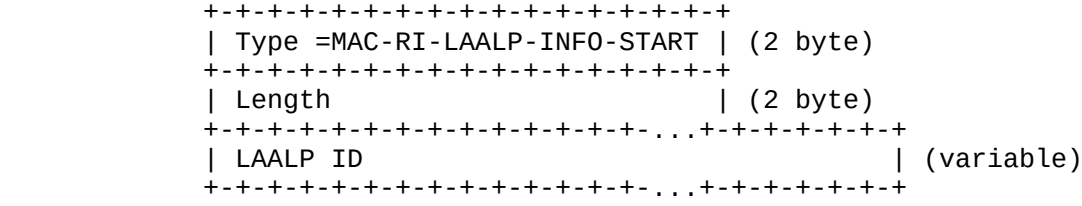
- LAALP ID (LAAP ID Size bytes): The ID of the LAALP.

This sub-TLV may occur multiple times with the same RBv pseudo-nickname with the meaning that all of the LAALPs listed are identified by that pseudo-nickname. For example, if there are LAALP IDs of different length, then the LAALP IDs of each size would have to be listed in a separate sub-TLV.

On receipt of such a sub-TLV, if RBn is not an LAALP related edge RBridge, it ignores the sub-TLV. Otherwise, if RBn is also a member RBridge of the RBv identified by the list of LAALPs, it associates the pseudo-nickname with the ports of these LAALPs and downloads the association onto data plane fast path logic.

9.3. MAC-RI-LAALP Boundary APPsub-TLVs

In this document, two APPsub-TLVs are used as boundary APPsub-TLVs for edge RBridge to enclose the MAC-RI TLV(s) containing the MAC address information leant form local port of an LAALP when this RBridge wants to share the information with other edge RBridges. They are defined as TRILL APPsub-TLVs [RFC7357]. The MAC-RI-LAALP-INFO-START APPsub-TLV has the following format:

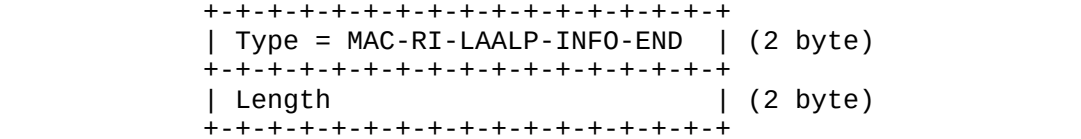


- MAC-RI-LAALP-INFO-START (2 bytes): Defines the type of this APPsub-TLV, #tbd3.

- Length (2 bytes): the size of the following LAALP ID. 8 if the LAALP listed is an MAC-LAG.

- LAALP ID (variable): The ID of the LAALP (for example, for an MC-LAG the ID as specified in Section 5.3.2 in [802.1AX]). This ID identifies the LAALP for all MAC addresses contained in following MAC-RI TLVs until an MAC-RI-LAALP-INFO-END APPsub-TLV is encountered.

MAC-RI-LAALP-INFO-END APPsub-TLV is defined as follows:



- MAC-RI-LAALP-INFO-END (2 bytes): Defines the type of this sub-TLV, #tbd4.

- Length (2 bytes): 0.

This pair of APPsub-TLVs can be carried multiple times in an ESADI LSP and in multiple ESADI-LSPs. When an LAALP related edge RBridge (say RBn) wants to share with other edge RBridges the MAC addresses learned on its local ports of different LAALPs, it uses one or more pairs of such APPsub-TLVs for each of such LAALPs in its ESADI-LSPs. Each encloses the MAC-RI TLVs containing the MAC addresses learned from a specific LAALP. Furthermore, if the LAALP is served by a local RBv, the value of Topology ID/Nickname field in the relative MAC-RI

TLVs SHOULD be the pseudo-nickname of the RBv rather than one of the RBn's regular nickname or zero. Then on receipt of such a MAC-RI TLV, remote RBridges know that the contained MAC addresses are reachable through the RBv.

On receipt of such boundary APPsub-TLVs, when the edge RBridge is not an LAALP related one or cannot recognize such sub-TLVs, it ignores them and continues to parse the enclosed MAC-RI TLVs per [RFC7357]. Otherwise, the recipient parses the boundary APPsub-TLVs, and

1) If the edge RBridge is configured with the contained LAALP and the LAALP is also enabled locally, it treats all the MAC addresses, contained in the following MC-RI TLVs enclosed by the corresponding pair of boundary APPsub-TLVs, as if they were learned from its local port of that LAALP;

2) Else, it ignores these boundary APPsub-TLVs and continues to parse the following MAC-RI TLVs per [RFC7357] until another pair of boundary APPsub-TLVs is encountered.

10. OAM Packets

Attention must be paid when generating OAM packets. To ensure the response messages can return to the originating member RBridge of an RBv, pseudo-nickname cannot be used as ingress nickname in TRILL OAM messages, except in the response to an OAM message that has that RBv's pseudo-nickname as egress nickname. For example, assume RB1 is a member RBridge of RBvi, RB1 cannot use RBvi's pseudo-nickname as the ingress nickname when originating OAM messages; otherwise the responses to the messages may be delivered to another member RBridge of RBvi rather than RB1. But when RB1 responds to the OAM message with RBvi's pseudo-nickname as egress nickname, it can use that pseudo-nickname as ingress nickname in the response message.

Since RBridges cannot use OAM messages for the learning of MAC addresses (Section 3.2.1 of [RFC7174]), it will not lead to MAC address flip-flopping at a remote RBridge even though RB1 uses its regular nicknames as ingress nicknames in its TRILL OAM messages while uses RBvi's pseudo-nickname in its TRILL Data packets.

11. Configuration Consistency

It is important that the VLAN membership of all the RBridge ports in an LAALP MUST be the same. Any inconsistencies in VLAN membership may result in packet loss or non-shortest paths.

Take Figure 1 for example, suppose RB1 configures VLAN1 and VLAN2 for

the link CE1-RB1, while RB2 only configures VLAN1 for the CE1-RB2 link. Both RB1 and RB2 use the same ingress nickname RBv for all frames originating from CE1. Hence, a remote RBridge RBx will learn that CE1's MAC address in VLAN2 is originating from RBv. As a result, on the returning path, remote RBridge RBx may deliver VLAN2 traffic to RB2. However, RB2 does not have VLAN2 configured on CE1-RB2 link and hence the frame may be dropped or has to be redirected to RB1 if RB2 knows RB1 can reach CE1 in VLAN2.

Furthermore, it is important that if any VLAN in an LAALP is being mapped by edge RBridges to an FGL [RFC7172], that the mapping MUST be same for all edge RBridge ports in the LAALP. Otherwise, for example, unicast FGL TRILL Data packets from remote RBridges may get mapped into different VLANs depending on which edge RBridge receives and egresses them.

12. Security Considerations

This draft does not introduce any extra security risks. For general TRILL Security Considerations, see [RFC6325]. For ESADI Security Considerations, see [RFC7357].

13. IANA Considerations

IANA is requested to allocate code points tbd1, tbd2, tbd3 and tbd4 from the range below 255 for the 4 TRILL APPsub-TLVs specified in Section 9 and add them to the TRILL APPsub-TLV Types registry.

14. Acknowledgments

We would like to thank Mingjiang Chen for his contributions to this document. Additionally, we would like to thank Erik Nordmark, Les Ginsberg, Ayan Banerjee, Dinesh Dutt, Anoop Ghanwani, Janardhanan Pathang, Jon Hudson and Fangwei Hu for their good questions and comments.

15. Contributing Authors

Weiguo Hao
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56623144
Email: haoweiguo@huawei.com

Donald E. Eastlake, III
Huawei Technologies
155 Beaver Street
Milford, MA 01757 USA

Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

16. References

16.1. Normative References

[CMT] T. Senevirathne, J. Pathangi, and J. Hudson, "Coordinated Multicast Trees (CMT) for TRILL", draft-ietf-trill-cmt-01.txt Work in Progress, April 2014.

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

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

[RFC6165] Banerjee, A. and D. Ward, "Extensions to IS-IS for Layer-2 Systems", RFC 6165, April 2011.

[RFC6325] R. Perlman, D. Eastlake, D. Dutt, S. Gai, and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

[RFC7172] Eastlake 3rd, D., Zhang, M., Agarwal, P., Perlman, R., and D. Dutt, "Transparent Interconnection of Lots of Links (TRILL): Fine-Grained Labeling", RFC 7172, May 2014.

[RFC7176] D. Eastlake, A. Banerjee, A. Ghanwani, and R. Perlman, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", RFC7176, May 2014.

[RFC7357] Zhai, H., Hu, F., Perlman, R., Eastlake 3rd, D., and O.

Stokes, "Transparent Interconnection of Lots of Links (TRILL): End Station Address Distribution Information (ESADI) Protocol", RFC 7357, September 2014.

[RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, September 2014.

[rfc7180bis] D. Eastlake, et al., draft-eastlake-trill-clear-correct, work in progress.

[802.1AX] IEEE, "IEEE Standard for Local and Metropolitan Area/ networks Link Aggregation", 802.1AX-2008, 1 January 2008.

16.2. Informative References

[RFC7379] Li, Y., Hao, W., Perlman, R., Hudson, J., and H. Zhai, "Problem Statement and Goals for Active-Active Connection at the Transparent Interconnection of Lots of Links (TRILL) Edge", RFC 7379, October 2014,.

Authors' Addresses

Hongjun Zhai
Jinling Institute of Technology
99 Hongjing Avenue, Jiangning District
Nanjing, Jiangsu 211169
China

Email: honjun.zhai@tom.com

Tissa Senevirathne
Cisco Systems
375 East Tasman Drive
San Jose, CA 95134
USA

Phone: +1-408-853-2291
Email: tsenevir@cisco.com

Radia Perlman
EMC
2010 256th Avenue NE, #200
Bellevue, WA 98007
USA

Email: Radia@alum.mit.edu

Mingui Zhang
Huawei Technologies
Huawei Building, No.156 Beiqing Rd.
Beijing, Beijing 100095
China

Email: zhangmingui@huawei.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56625409
Email: liyizhou@huawei.com

TRILL
Internet Draft
Intended status: Standards Track

Tissa Senevirathne
Norman Finn
Deepak Kumar
Samer Salam
Cisco

Liang Xia
Weiguo Hao
Huawei

September 9, 2014

Expires: March 2015

YANG Data Model for TRILL Operations, Administration, and Maintenance (OAM)
draft-ietf-trill-yang-oam-00.txt

Status of this Memo

Copyright Notice

Abstract

This document presents YANG Data model for TRILL OAM. It extends the Generic YANG model for OAM defined in [GENYANGOAM] with TRILL technology specifics.

Table of Contents

1. Introduction

Fault Management for TRILL is defined in [TRILLOAMFM]. TRILL Fault Management utilizes the [8021Q] CFM model and extends CFM with technology specific details. Those technology specific extensions are flow-entropy for multipath support, MEP addressing on TRILL identifiers, and so on. The extensions are explained in detail in [TRILLOAMFM]. In this document, we extend the YANG model defined in [GENYANGOAM] with TRILL OAM specifics.

2. Conventions used in this document

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].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

2.1. Terminology

CCM - Continuity Check Message [8021Q]

ECMP - Equal Cost Multipath

LBM - Loopback Message [8021Q]

MP - Maintenance Point [8021Q]

MEP - Maintenance End Point [TRLOAMFRM] [8021Q] [RFC6371]

MIP - Maintenance Intermediate Point [TRLOAMFRM] [8021Q] [RFC6371]

MA - Maintenance Association [8021Q] [TRLOAMFRM]

MD - Maintenance Domain [8021Q]

MTV - Multi-destination Tree Verification Message

OAM - Operations, Administration, and Maintenance [RFC6291]

TRILL - Transparent Interconnection of Lots of Links [RFC6325]

3. Architecture of OAM YANG Model and Relationship to TRILL OAM

The Generic OAM YANG model acts as the basis for other OAM YANG models. This allows users to traverse between OAM tools of different technologies at ease through a uniform API set. This is also referred as the nested OAM workflow. The following Figure depicts the relationship of TRILL OAM YANG model to Generic YANG Model.

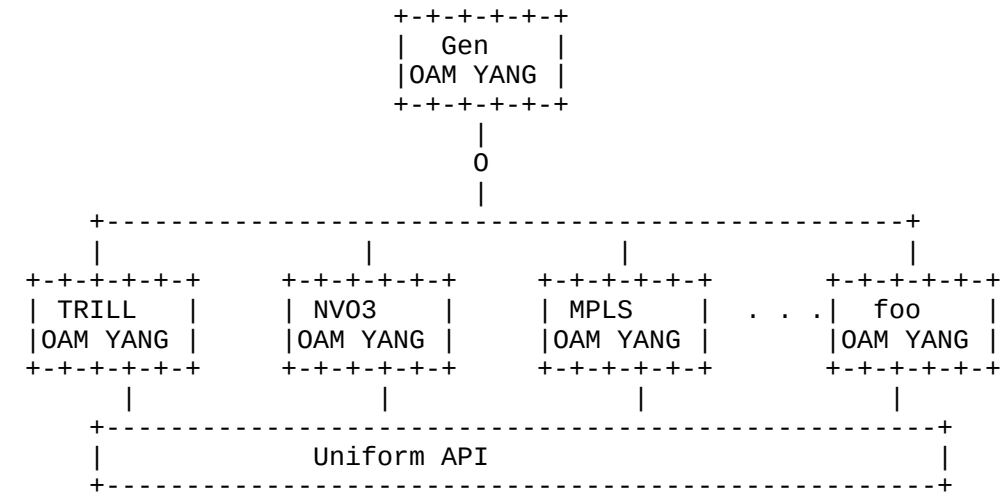


Figure 1 Relationship of TRILL OAM YANG model to Generic OAM YANG model

4. TRILL extensions to Generic YANG Model

The Technology parameter is defined in the [GENYANGOAM] as an identity. This allows easy extension of the YANG model by other technologies. Technology-specific extensions are applied only when the Technology parameter is set to the specific type. "trill" is defined as an identity that augments the base "technology-types".

```
identity trill {
    base goam:technology-types;
    description
      "trill type";
  }
```

Figure 2 Trill identity type.

4.1. MEP address

In TRILL, the MEP address is the 2 octet RBridge Nickname. In [GENYANGOAM] MEP address is defined as a combination choice and case statement. We augment this to include TRILL RBridge nickname.

```
augment
"/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:mep-
address" {
    case mep-address-trill {
      leaf mep-address-trill {
        when "/goam:domains/goam:domain/goam:technology='trill'";
        type tril-rb-nickname;
      }
    }
  }
```

Figure 3 Augment MEP address

4.2. Flow-entropy

In TRILL, flow-entropy is defined as a 96 octet field. [GENYANGOAM] defines a placeholder for flow-entropy. This allows other technologies to easily augment that to include technology-specific augmentations. Below figure depicts an example of augmenting flow-entropy to include TRILL flow-entropy.

```
augment "/goam:domains/goam:domain/goam:MAs/goam:MA/goam:flow-
entropy" {
    case flow-entropy-trill {
      leaf flow-entropy-trill {
        type flow-entropy-trill;
      }
    }
  }
```

Figure 4 TRILL flow-entropy

4.3. Context-id

In TRILL, context-id is either 12 bit VLAN identifier or 24 bit fine-grained label. [GENYANGOAM] defines a placeholder for context-id. This allows other technologies to easily augment that to include technology specific extensions. The snippet below depicts an example of augmenting context-id to include the TRILL context-id.

```
augment
"/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:context-id"
{
    case context-id-vlan {
      leaf context-id-vlan {
        type vlan;
      }
    }
    case context-id-fgl {
      leaf context-id-fgl {
        type fgl;
      }
    }
  }
```

Figure 5 TRILL context-id

4.4. rpc definitions

The rpc model facilitates issuing commands to a NETCONF server (in this case to the device that needs to execute the OAM command) and obtaining a response. Grouping statement command-ext-trill defines the input extensions for TRILL.

Multicast Tree Verification (MTV) [TRILLOAMFM] rpc command, defined in TRILL YANG model, is TRILL specific and allows to verify connectivity as well as data-plane and control-plane integrity of TRILL multicast forwarding.

5. OAM data hierarchy

The complete data hierarchy related to the OAM YANG model is presented below. The following notations are used within the data tree and carry the meaning as noted below.

Each node is printed as:

<status> <flags> <name> <opts> <type>

<status> is one of:
- \+ for current
- x for deprecated
- o for obsolete

<flags> is one of:

rw for configuration data
ro for non-configuration data
-x for rpcs
-n for notifications

<name> is the name of the node

If the node is augmented into the tree from another module, its name is printed as <prefix>:<name>.

<opts> is one of:

? for an optional leaf or choice
! for a presence container
* for a leaf-list or list
[<keys>] for a list’s keys

<type> is the name of the type for leafs and leaf-lists

```
module: trill-oam
augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:mep-
address:
   +--:(mep-address-trill)
      +--rw mep-address-trill?   tril-rb-nickname
augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:context-id:
   +--:(context-id-vlan)
   |  +--rw context-id-vlan?   vlan
   +--:(context-id-fgl)
      +--rw context-id-fgl?    fgl
augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:flow-entropy:
   +--:(flow-entropy-trill)
      +--rw flow-entropy-trill?   flow-entropy-trill
augment
/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:context-id:
   +--:(context-id-vlan)
   |  +--rw context-id-vlan?   vlan
   +--:(context-id-fgl)
      +--rw context-id-fgl?    fgl
augment
/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:flow-
entropy:
   +--:(flow-entropy-trill)
      +--rw flow-entropy-trill?   flow-entropy-trill
augment
/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:session/goam
:context-id:
   +--:(context-id-vlan)
   |  +--rw context-id-vlan?   vlan
   +--:(context-id-fgl)
      +--rw context-id-fgl?    fgl
augment
/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:session/goam
:flow-entropy:
   +--:(flow-entropy-trill)
      +--rw flow-entropy-trill?   flow-entropy-trill
augment /goam:ping/goam:input:
   +--ro (out-of-band)?
   |  +--:(ipv4-address)
   |  |  +--ro ipv4-address?      inet:ipv4-address
   |  +--:(ipv6-address)
   |  |  +--ro ipv6-address?      inet:ipv6-address
   |  +--:(trill-nickname)
   |     +--ro trill-nickname?    tril-rb-nickname
   +--ro diagnostic-vlan?   boolean
augment /goam:ping/goam:input/goam:context-id:
```

```
      +--:(context-id-vlan)
      |  +--ro context-id-vlan?   vlan
      +--:(context-id-fgl)
         +--ro context-id-fgl?    fgl
   augment /goam:ping/goam:input/goam:flow-entropy:
      +--:(flow-entropy-trill)
         +--ro flow-entropy-trill?   flow-entropy-trill
   augment /goam:ping/goam:input/goam:source-mep/goam:mep-address:
      +--:(trill-nickname)
         +--ro trill-nickname?   tril-rb-nickname
   augment /goam:ping/goam:input/goam:destination-mep/goam:mep-address:
      +--:(trill-nickname)
         +--ro trill-nickname?   tril-rb-nickname
   augment /goam:ping/goam:output:
      +--ro upstream-rbridge?   tril-rb-nickname
      +--ro next-hop-rbridge*   tril-rb-nickname
   augment /goam:trace-route/goam:input:
      +--ro (out-of-band)?
      |  +--:(ipv4-address)
      |  |  +--ro ipv4-address?      inet:ipv4-address
      |  +--:(ipv6-address)
      |  |  +--ro ipv6-address?      inet:ipv6-address
      |  +--:(trill-nickname)
      |     +--ro trill-nickname?    tril-rb-nickname
      +--ro diagnostic-vlan?   boolean
   augment /goam:trace-route/goam:input/goam:context-id:
      +--:(context-id-vlan)
      |  +--ro context-id-vlan?   vlan
      +--:(context-id-fgl)
         +--ro context-id-fgl?    fgl
   augment /goam:trace-route/goam:input/goam:flow-entropy:
      +--:(flow-entropy-trill)
         +--ro flow-entropy-trill?   flow-entropy-trill
   augment /goam:trace-route/goam:input/goam:source-mep/goam:mep-
   address:
      +--:(trill-nickname)
         +--ro trill-nickname?   tril-rb-nickname
   augment /goam:trace-route/goam:input/goam:destination-mep/goam:mep-
   address:
      +--:(trill-nickname)
         +--ro trill-nickname?   tril-rb-nickname
   augment /goam:trace-route/goam:output/goam:response/goam:destination-
   mep/goam:mep-address:
      +--:(trill-nickname)
         +--ro trill-nickname?   tril-rb-nickname
   augment /goam:trace-route/goam:output/goam:response:
      +--ro upstream-rbridge?   tril-rb-nickname
```

```
      +--ro next-hop-rbridge*   tril-rb-nickname
   rpcs:
      +---x mtv
         +--ro input
         |  +--ro technology             identityref
         |  +--ro md-name-format         MD-name-format
         |  +--ro md-name?               binary
         |  +--ro md-level               int32
         |  +--ro ma-name-format         MA-name-format
         |  +--ro ma-name                binary
         |  +--ro (out-of-band)?
         |  |  +--:(ipv4-address)
         |  |  |  +--ro ipv4-address?          inet:ipv4-address
         |  |  +--:(ipv6-address)
         |  |  |  +--ro ipv6-address?          inet:ipv6-address
         |  |  +--:(trill-nickname)
         |  |     +--ro trill-nickname?        tril-rb-nickname
         |  +--ro diagnostic-vlan?       boolean
         |  +--ro (context-id)?
         |  |  +--:(context-id-vlan)
         |  |  |  +--ro context-id-vlan?       vlan
         |  |  +--:(context-id-fgl)
         |  |     +--ro context-id-fgl?        fgl
         |  +--ro (flow-entropy)?
         |  |  +--:(flow-entropy-null)
         |  |  |  +--ro flow-entropy-null?     empty
         |  |  +--:(flow-entropy-trill)
         |  |     +--ro flow-entropy-trill?    flow-entropy-trill
         |  +--ro max-hop-count?         uint8
         |  +--ro type?                  identityref
         |  +--ro scope*                 tril-rb-nickname
         |  +--ro ecmp-choice?           goam:ecmp-choices
         |  +--ro outgoing-interfaces* [interface]
         |  |  +--ro interface    if:interface-ref
         |  +--ro source-mep
         |  |  +--ro (mep-address)?
         |  |  |  +--:(mac-address)
         |  |  |  |  +--ro mac-address?    yang:mac-address
         |  |  |  +--:(ipv4-address)
         |  |  |  |  +--ro ipv4-address?   inet:ipv4-address
         |  |  |  +--:(ipv6-address)
         |  |  |     +--ro ipv6-address?   inet:ipv6-address
         |  |  +--ro mep-id?         goam:MEP-id
         |  +--ro destination-mep
         |     +--ro (mep-address)?
         |     |  +--:(mac-address)
         |     |  |  +--ro mac-address?    yang:mac-address
```

```
|     |  +--:(ipv4-address)
|     |  |  +--ro ipv4-address?   inet:ipv4-address
|     |  +--:(ipv6-address)
|     |     +--ro ipv6-address?   inet:ipv6-address
|     +--ro mep-id?         goam:MEP-id
+--ro output
   +--ro response* [mep-address mep-id]
      +--ro hop-count?                  uint8
      +--ro mep-id                      goam:MEP-id
      +--ro mep-address                 tril-rb-nickname
      +--ro next-hop-rbridge*           tril-rb-nickname
      +--ro upstream-rbridge?           tril-rb-nickname
      +--ro multicast-receiver-count?   uint32
      +--ro tx-packt-count?             oam-counter32
      +--ro rx-packet-count?            oam-counter32
      +--ro min-delay?                  oam-counter32
      +--ro average-delay?              oam-counter32
      +--ro max-delay?                  oam-counter32
```

Figure 1 Data hierarchy of TRILL OAM

6. OAM YANG module

<CODE BEGINS> file "xxx.yang"

```
module trill-oam {
  namespace "urn:cisco:params:xml:ns:yang:tril-oam";
  prefix trilloam;

  import gen-oam {
    prefix goam;
  }
  import ietf-inet-types {
    prefix inet;
  }
  import ietf-interfaces {
    prefix if;
  }

  revision 2014-04-16 {
    description
      "Initial revision.";
  }

 identity trill {
    base goam:technology-types;
    description
      "trill type";
  }

  typedef tril-rb-nickname {
    type uint16;
  }

  typedef flow-entropy-trill {
    type binary {
      length "1..96";
    }
  }

  typedef vlan {
    type uint16 {
      range "0..4095";
    }
  }

  typedef fgl {
    type uint32;
```

```
}

identity trill-mtv {
  base goam:command-sub-type;
  description
    "identfies this command as multicast tree verification comand";
}

identity trill-ping {
  base goam:command-sub-type;
}

identity trill-trace-route {
  base goam:command-sub-type;
}

grouping command-ext-trill {
  description
    "group the rpc command extensions for trill";
  choice out-of-band {
    case ipv4-address {
      leaf ipv4-address {
        type inet:ipv4-address;
      }
    }
    case ipv6-address {
      leaf ipv6-address {
        type inet:ipv6-address;
      }
    }
    case trill-nickname {
      leaf trill-nickname {
        type tril-rb-nickname;
      }
    }
    description
      "presence of this node indicate out of band request needed";
  }
  leaf diagnostic-vlan {
    type boolean;
    description
      "indicates whether to include diagnostic VLAN/fgl TLV or not
       in the request.  actual value is the VLAN/FGL specified
       in the command";
  }
}
```

```
    augment
  "/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:mep-
  address" {
      case mep-address-trill {
        leaf mep-address-trill {
          when "/goam:domains/goam:domain/goam:technology='trill'";
          type tril-rb-nickname;
        }
      }
    }
    augment "/goam:domains/goam:domain/goam:MAs/goam:MA/goam:context-
  id" {
      case context-id-vlan {
        leaf context-id-vlan {
          type vlan;
        }
      }
      case context-id-fgl {
        leaf context-id-fgl {
          type fgl;
        }
      }
    }
    augment "/goam:domains/goam:domain/goam:MAs/goam:MA/goam:flow-
  entropy" {
      case flow-entropy-trill {
        leaf flow-entropy-trill {
          type flow-entropy-trill;
        }
      }
    }
    augment
  "/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:context-id"
  {
      case context-id-vlan {
        leaf context-id-vlan {
          type vlan;
        }
      }
      case context-id-fgl {
        leaf context-id-fgl {
          type fgl;
        }
      }
    }
```

```
    augment
  "/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:flow-
  entropy" {
      case flow-entropy-trill {
        leaf flow-entropy-trill {
          type flow-entropy-trill;
        }
      }
    }
    augment
  "/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:session/goa
  m:context-id" {
      case context-id-vlan {
        leaf context-id-vlan {
          type vlan;
        }
      }
      case context-id-fgl {
        leaf context-id-fgl {
          type fgl;
        }
      }
    }
    augment
  "/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:session/goa
  m:flow-entropy" {
      case flow-entropy-trill {
        leaf flow-entropy-trill {
          type flow-entropy-trill;
        }
      }
    }
    augment "/goam:ping/goam:input" {
      uses command-ext-trill;
    }
    augment "/goam:ping/goam:input/goam:context-id" {
      case context-id-vlan {
        leaf context-id-vlan {
          type vlan;
        }
      }
      case context-id-fgl {
        leaf context-id-fgl {
          type fgl;
        }
      }
    }
```

```
    augment "/goam:ping/goam:input/goam:flow-entropy" {
      case flow-entropy-trill {
        leaf flow-entropy-trill {
          type flow-entropy-trill;
        }
      }
    }
    augment "/goam:ping/goam:input/goam:source-mep/goam:mep-address" {
      case trill-nickname {
        leaf trill-nickname {
          type tril-rb-nickname;
        }
      }
    }
    augment "/goam:ping/goam:input/goam:destination-mep/goam:mep-
   address" {
      case trill-nickname {
        leaf trill-nickname {
          type tril-rb-nickname;
        }
      }
    }
    augment "/goam:ping/goam:output" {
      description
        "adds trill specific items on the response";
      leaf upstream-rbridge {
        type tril-rb-nickname;
      }
      leaf-list next-hop-rbridge {
        type tril-rb-nickname;
        description
          "nickname of the next hop RBRdige";
      }
    }
    augment "/goam:trace-route/goam:input" {
      uses command-ext-trill;
    }
    augment "/goam:trace-route/goam:input/goam:context-id" {
      case context-id-vlan {
        leaf context-id-vlan {
          type vlan;
        }
      }
      case context-id-fgl {
        leaf context-id-fgl {
          type fgl;
        }
```

```
      }
    }
    augment "/goam:trace-route/goam:input/goam:flow-entropy" {
      case flow-entropy-trill {
        leaf flow-entropy-trill {
          type flow-entropy-trill;
        }
      }
    }
    augment "/goam:trace-route/goam:input/goam:source-mep/goam:mep-
  address" {
      case trill-nickname {
        leaf trill-nickname {
          type tril-rb-nickname;
        }
      }
    }
    augment "/goam:trace-route/goam:input/goam:destination-
  mep/goam:mep-address" {
      case trill-nickname {
        leaf trill-nickname {
          type tril-rb-nickname;
        }
      }
    }
    augment "/goam:trace-
  route/goam:output/goam:response/goam:destination-mep/goam:mep-
  address" {
      case trill-nickname {
        leaf trill-nickname {
          type tril-rb-nickname;
        }
      }
    }
    augment "/goam:trace-route/goam:output/goam:response" {
      description
        "adds trill specific items on the response";
      leaf upstream-rbridge {
        type tril-rb-nickname;
      }
      leaf-list next-hop-rbridge {
        type tril-rb-nickname;
        description
          "nickname of the next hop RBRdige";
      }
    }
    rpc mtv {
```

```
      description
        "Generates Trace-route and return response. Starts with TTL
         of one and increment by one at each hop. Untill destination
         reached or TTL reach max valune";
      input {
        uses goam:maintenance-domain {
          description
            "Specifies the MA-domain";
        }
        uses goam:ma-identifier {
          description
            "identfies the Maintenance association";
        }
        uses command-ext-trill {
          description
            "defines extensions needed for trill.
             We are using this structure so mtv command is in line
             with ping and trace-route";
        }
        choice context-id {
          case context-id-vlan {
            leaf context-id-vlan {
              type vlan;
            }
          }
          case context-id-fgl {
            leaf context-id-fgl {
              type fgl;
            }
          }
        }
        choice flow-entropy {
          case flow-entropy-null {
            leaf flow-entropy-null {
              type empty;
            }
          }
          case flow-entropy-trill {
            leaf flow-entropy-trill {
              type flow-entropy-trill;
            }
          }
        }
        leaf max-hop-count {
          type uint8;
          default "255";
          description
```

```
            "Defines maximum value of hop count";
          }
          leaf type {
            type identityref {
              base goam:command-sub-type;
            }
            description
              "defines different command types";
          }
          leaf-list scope {
            type tril-rb-nickname;
            reference "draft-ietf-trill-oam-fm";
            description
              "This list contain rbridges that needed to respond
               Empty list indicate all Rbridges needed to respond";
          }
          leaf ecmp-choice {
            type goam:ecmp-choices;
            description
              "0 means use the specified interface
               1 means use round robin";
          }
          list outgoing-interfaces {
            key "interface";
            leaf interface {
              type if:interface-ref;
            }
          }
          container source-mep {
            uses goam:mep-address;
            leaf mep-id {
              type goam:MEP-id;
            }
          }
          container destination-mep {
            uses goam:mep-address;
            leaf mep-id {
              type goam:MEP-id;
            }
          }
        }
        output {
          list response {
            key "mep-address mep-id";
            leaf hop-count {
              type uint8;
            }
```

```
            leaf mep-id {
              type goam:MEP-id;
            }
            leaf mep-address {
              type tril-rb-nickname;
            }
            leaf-list next-hop-rbridge {
              type tril-rb-nickname;
              description
                "list of downstream rbridges. There is no perticular
   order";
            }
            leaf upstream-rbridge {
              type tril-rb-nickname;
            }
            leaf multicast-receiver-count {
              type uint32;
              description
                "number of ports that are interested in this multicast
   stream";
            }
            uses goam:monitor-stats;
          }
        }
      }
   }


   <CODE ENDS>
```

Figure 2 YANG module of OAM

7. Base Mode for TRILL OAM

The Base Mode defines default configuration that MUST be present in the devices that comply with this document. Base Mode allows users to have a zero-touch experience. Details of TRILL Base Mode for OAM are defined in [TRILLOAMFM].

8. Security Considerations

TBD

9. IANA Considerations

This document registers the following namespace URI in the IETF XML registry.

URI:TBD

10. References

10.1. Normative References

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

[RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

[8021Q] IEEE, "Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q-2011, August, 2011.

10.2. Informative References

[Y1731] ITU, "OAM functions and mechanisms for Ethernet based networks", ITU-T G.8013/Y.1731, July, 2011.

[TRLOAMFRM] Salam, S., et.al., "TRILL OAM Framework", draft-ietf-trill-oam-framework, Work in Progress, November, 2012.

[RFC6291] Andersson, L., et.al., "Guidelines for the use of the "OAM" Acronym in the IETF" RFC 6291, June 2011.

[RFC6325] Perlman, R., et.al., "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

[GENYANGOAM] Senevirathne, T., et.al., "YANG Data Model for Operations, Administration and Maintenance (OAM)", Work in Progress, March, 2014.

11. Acknowledgments

Giles Heron came up with the idea of developing a YANG model as a way of creating a unified OAM API set (interface), work in this document is largely an inspiration of that. Alexander Clemm provided many valuable tips, comments and remarks that helped to refine the YANG model presented in this document.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Tissa Senevirathne
CISCO Systems
375 East Tasman Drive.
San Jose, CA 95134
USA.

Phone: 408-853-2291
Email: tsenevir@cisco.com

Norman Finn
CISCO Systems
510 McCarthy Blvd
Milpitas, CA 95035.

Email: nfinn@cisco.com

Deepak Kumar
CISCO Systems
510 McCarthy Blvd
Milpitas, CA 95035.

Email: dekumar@cisco.com

Samer Salam
CISCO Systems
595 Burrard St. Suite 2123
Vancouver, BC V7X 1J1, Canada

Email: ssalam@cisco.com

Liang Xia
Huawei technologies

Email: frank.xialiang@huawei.com

Weiguo Hao
Huawei Technologies
101 Software Avenue
Nanjing 210012, China

Email: haoweiguo@huawei.com

TRILL WG
Internet-Draft
Intended status: Standards Track
Expires: April 20, 2015

Radia. Perlman
Intel Labs
Fangwei. Hu
ZTE Corporation
Donald. Eastlake 3rd
Huawei technology
Kesava. Krupakaran
Dell
Ting. Liao
ZTE Corporation

October 17, 2014

TRILL Smart Endnodes
draft-perlman-trill-smart-endnodes-04.txt

Abstract

This draft addresses the problem of the size and freshness of the endnode learning table in edge RBridges, by allowing endnodes to volunteer for endnode learning and encapsulation/decapsulation. Such an endnode is known as a "smart endnode". Only the attached RBridge can distinguish a "smart endnode" from a "normal endnode". The smart endnode uses the nickname of the attached RBridge, so this solution does not consume extra nicknames. The solution also enables Fine Grained Label aware endnodes.

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 April 20, 2015.

Table of Contents

1. Introduction

The IETF TRILL (Transparent Interconnection of Lots of Links) protocol [RFC6325] provides optimal pair-wise data frame forwarding without configuration, safe forwarding even during periods of temporary loops, and support for multipathing of both unicast and multicast traffic. TRILL accomplishes this by using IS-IS [IS-IS] [RFC7176] link state routing and encapsulating traffic using a header that includes a hop count. Devices that implement TRILL are called "RBridges" (Routing Bridges) or TRILL Switches.

An RBridge that attaches to endnodes is called an "edge RBridge", whereas one that exclusively forwards encapsulated frames is known as a "transit RBridge". An edge RBridge traditionally is the one that encapsulates a native Ethernet packet with a TRILL header, or that receives a TRILL-encapsulated packet and decapsulates the TRILL

header. To encapsulate efficiently, the edge RBridge must keep an "endnode table" consisting of (MAC,Data Label, TRILL egress switch nickname) sets, for those remote MAC addresses in Data Labels currently communicating with endnodes to which the edge RBridge is attached.

These table entries might be configured, received from ESADI [RFC7357], looked up in a directory [RFC7067], or learned from decapsulating received traffic. If the edge RBridge has attached endnodes communicating with many remote endnodes, this table could become large. Also, if one of the MAC addresses and Data Labels in the table has moved to a different remote TRILL switch, it might be difficult for the edge RBridge to notice this quickly, and because the edge RBridge is encapsulting to the incorrect egress RBridge, the traffic will get lost.

For these reasons, it is desirable for an endnode E (whether it is a server, hypervisor, or VM) to maintain the endnode table for remote endnodes that E is corresponding with. This eliminates the need for the edge RBridge RBx, to which E is connected, to know about those nodes (unless some non-smart endnode attached to RBx is also corresponding with those nodes), Once D is unreachable for E, which could be determined through ICMP messages or other techniques, the smart endnode should delete the entry of (MAC, Data Label, nickname). If D moves to a new place, E should attempt to acquire a fresh entry for D by flooding to D, examining updates to the ESADI link state database,or consulting a directory.

The mechanism in this draft is that E issue a Smart-Hello (even though E is just an endnode), indicating E’s desire to act as a smart endnode, together with the set of MAC addresses and Data Labels that E owns, and whether E would like to receive ESADI packets. E learns from RBx’s Smart-Hello, whether RBx is capable of having a smart endnode neighbor, what RBx’s nickname is, and which trees RBx can use when RBx ingresses multi-destination frames. Although E transmits Smart-Hellos, E does not transmit or receive LSPs or E-L1FS FS-LSPs[I-D.eastlake-trill-rfc7180bis].

RBx will accept already-encapsulated TRILL Data packets from E (perhaps verifying that the source MAC and Data Label is indeed one of the ones that E owns, that the ingress RBridge field is RBx’s, and if the packet is an encapsulated multi-destination frame, the tree selected is one of the ones that RBx has claimed it will choose). When RBx receives (from the campus) a TRILL Data packet with RBx’s nickname as egress, RBx checks whether the destination MAC address and Data Label in the inner packet is one of the MAC addresses and Data Labels that E owns, and if so, RBx forwards the packet onto E’s port, keeping it encapsulated.

Since a smart endnode can encapsulate TRILL Data frames, it can cause the Inner.Lable to be a Fine Grained Label [RFC7172], thus this method supports FGL aware endnodes.

2. Terminology

Edge RBridge: An RBridge providing endnode service on at least one of its ports.

Data Label: VLAN or FGL.

ESADI: End Station Address Distribution Information [RFC7357].

FGL: Fine Grained Label [RFC7172].

IS-IS: Intermediate System to Intermediate System [IS-IS].

RBridge: Routing Bridge, an alternative name for a TRILL switch.

Smart endnode: An endnode that has the capability specified in this document including learning and maintaining(MAC, Data Lable, Nickname) entries and encapsulating/decapsulating TRILL frame.

Transit RBridge: An RBridge exclusively forwards encapsulated frames.

TRILL: Transparent Interconnection of Lots of Links [RFC6325].

TRILL switch: a device the implements the TRILL protocol; an alternative term for an RBridge.

3. Smart-Hello Content

Suppose endnode E is attached to RBridge RBx. In order for E to act as a smart endnode, both E and RBx have to be signaled. The logical choice of frame to do this is Smart-Hello.

3.1. Edge RBridge’s Smart-Hello

For smart endnode operation, RBx’s Smart-Hello must contain the following information:

- RBridge’s nickname. The nickname sub-TLV (Specified in section 2.3.2 in [RFC7176])could be reused here, and TLV 242 (IS-IS router capability) should be updated to be carried in Smart-Hello frame.

- Tree that RBx can use when ingressing multi-destination frames. The Tree Identifiers Sub-TLV (Specified in section 2.3.4 in [RFC7176]) could be reused here.

- Smart endnode neighbor list. The TRILL Neighbor TLV (Specified in section 2.5 in [RFC7176]) could be reused.

3.2. Smart Endnode's Smart-Hello

A new TLV (S-MAC TLV)is defined for smart endnode. If there are several VLANs for that smart endnode, the TLV could be filled several times in smart endnode's Smart-Hello.

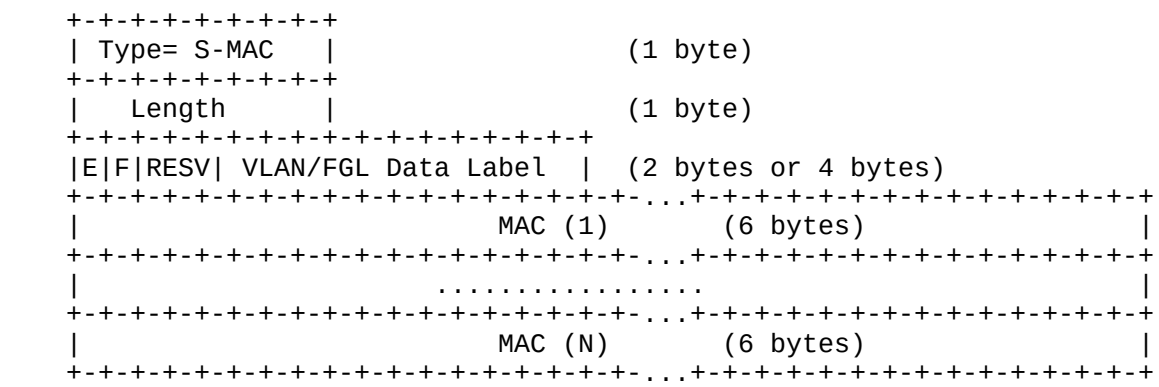


Figure 1 S-MAC TLV

- Type: S-MAC, the value is TBD.

- Length: Total number of bytes contained in the value field.

- E: one bit. If it sets to 1, which indicates that the endnode should receive ESADI frames.

- F: one bit. If it sets to 1, which indicates that the endnode supports FGL data label, otherwise, the VLAN/FGL Data Label [RFC7172] field is the VLAN ID.

- RESV: 2 bits or 6 bits, is reserved for the future use. If VLAN/FGL Data Label indicates the VLAN ID(or F flag sets to 0), the RESV field is 2 bits length, otherwise it is 6 bits.

- VLAN/FGL Data Label: This carries a 12-bits VLAN identifier or 24-bits FGL Data Label that is valid for all subsequent MAC addresses in this TLV, or the value zero if no VLAN/FGL data label is specified.

- MAC(i): This is the 48-bit MAC address reachable in the Data Label given from the IS that is announcing this TLV.

4. Frame Processing

4.1. Frame Processing for Smart Endnode

Smart endnode E does not issue or receive LSPs or E-L1FS FS-LSPs or calculate topology. E does the following:

- E maintains an endnode table of (MAC, Data Label, nickname) entries of end nodes with which the smart endnode is communicating. Entries in this table are populated the same way that an edge RBridge populates the entries in its table:
  - learning from (source, ingress) on packets it decapsulates.
  - from ESADI[RFC7357].
  - by querying a directory [RFC7067].
  - by having some entries configured.
- When E wishes to transmit to unicast destination D, if (D, nickname) is in E's endnode table, E encapsulates with ingress nickname=RBx, egress nickname as indicated in D's table entry. If D is unknown, D either queries a directory or encapsulates the packet as a multi-destination frame, using one of the trees that RBx has specified in RBx's Smart-Hello.
- When E wishes to transmit to a multicast or broadcast destination, E encapsulates the packet using one of the trees that RBx has specified.

The smart endnode E need not send Smart-Hellos as frequently as normal RBridges. These Smart-Hellos could be periodically unicast to the Appointed Forwarder RBx. In case RBx crashes and restarts, or the DRB changes, and E receives the Smart-Hello without mentioning E, then E SHOULD send a Smart-Hello immediately. If RBx is AF for any of the VLANs that E claims, RBx MUST list E in its Smart-Hellos as a smart endnode neighbor.

4.2. Frame Processing for Edge RBridge

The attached RBridge RBx does the following:

- If receiving an encapsulated unicast data frame from a port with a smart endnode, with RBx's nickname as ingress, RBx forwards the frame to the specified egress nickname, as with any encapsulated frame. However, RBx MAY filter the encapsulation frame based on the inner source MAC and Data Label as specified for the smart

endnode. If the MAC (or Data Label) are not among the expected set of the smart endnode, the frame would be dropped by the edge RBridge.

- o If receiving an mulit-destination TRILL Data packet from a port with smart endnode, RBridge RBx forwards the TRILL encapsulation to the TRILL campus based on the distribution tree. If there are some normal endnodes (i.e, non-smart endnode) attached to RBridge RBx, RBx should decapsulates the frame and sends the native frame to these ports.

- o When RBx receives a multicast frame from a remote RBridge, and the exit ports includes hybrid endnodes, it should send two copies of mulicast frames, one as native and the other as TRILL encapsulated frame. When smart endnode receives the encapsulated frame, it learns the remote (MAC, Data Label, Nickname) set, A smart endnodes ignores any native data frames. The normal endnode receives the native frame and learns the remote MAC address and ignore the native frame. This transit solution may bring some complex for the edge RBridge and waste network bandwidth, so it is recommended to avoid the hybrid endnodes scenario by attaching the smart endnodes and non-smart endnodes to different ports when deployed. Another solution is that if there are one or more endnodes on a link, the non-smart endnodes are ignored on a link; but we can configure a port to support mixed links. The RBx only sends TRILL encapsulated frame to the link in this situation.

5. Multi-homing

Now suppose E is attached to the TRILL campus in two places: to RBridges RB1 and RB2. There are two ways for this to work:

(1) E can choose either RB1 or RB2’s nickname, when encapsulating a frame, whether the encapsulated frame is sent via RB1 or RB2. If E wants to do active-active load splitting, and uses RB1’s nickname when forwarding through RB1, and RB2’s nickname when forwarding through RB2, which will cause the flip-floping of the endnode table entry in the remote RBridges (or smart endnodes). One solution is to set a multi-homing bit in the RESV field of the TRILL data Frame. When remote RBs or smart endnodes receive the data frame with the multi-homed bit set, the MAC entry (E, RB1’s nickname) and (E, RB2’s nickname) will be coexist as two entries for that MAC address. Another solution is to extend the ESADI protocol to distribute multiple attachments of a MAC address of a multi-homing group.(Please refer to the option C in section 4 of [I-D.ietf-trill-aa-multi-attach] for details).

(2) RB1 and RB2 might indicate, in their Smart-Hello, a virtual nickname that attached end nodes may use if they are multihomed to RB1 and RB2, separate from RB1 and RB2's nicknames (which they would also list in their Smart-Hello). This would be useful if there were many end nodes multihomed to the same set of RBridges. This would be analogous to a pseudonode nickname; return traffic would go via the shortest path from the source to the endnode, whether it is RB1 or RB2. If E loses connectivity to RB2, then E would revert to using RB1's nickname. In order to avoid RPF check issue for multi-destination frame, the affinity TLV [I-D.ietf-trill-cmt] is recommended to be used in this solution.

6. Security Considerations

For general TRILL Security Considerations, see [RFC6325].

7. Acknowledgements

8. IANA Considerations

IANA is requested to allocate a S-MAC TLV identifier. TLV 242(ISIS router capability) is required to updated to be carried by Smart-Hello frame.

9. Normative References

[I-D.eastlake-trill-rfc7180bis]
Eastlake, D., Zhang, M., Perlman, R., Banerjee, A., Ghanwani, A., and S. Gupta, "TRILL: Clarifications, Corrections, and Updates", draft-eastlake-trill-rfc7180bis-00 (work in progress), October 2014.

[I-D.ietf-trill-aa-multi-attach]
Zhang, M., Perlman, R., Corporation, Z., Durrani, D., Shaikh, M., and S. Gupta, "TRILL Active-Active Edge Using Multiple MAC Attachments", draft-ietf-trill-aa-multi-attach-01 (work in progress), August 2014.

[I-D.ietf-trill-cmt]
Senevirathne, T., Pathangi, J., and J. Hudson, "Coordinated Multicast Trees (CMT) for TRILL", draft-ietf-trill-cmt-04 (work in progress), October 2014.

[IS-IS] ISO/IEC 10589:2002, Second Edition,, "Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)", 2002.

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

[RFC6165] Banerjee, A. and D. Ward, "Extensions to IS-IS for Layer-2 Systems", RFC 6165, April 2011.

[RFC6325] Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

[RFC7067] Dunbar, L., Eastlake, D., Perlman, R., and I. Gashinsky, "Directory Assistance Problem and High-Level Design Proposal", RFC 7067, November 2013.

[RFC7172] Eastlake, D., Zhang, M., Agarwal, P., Perlman, R., and D. Dutt, "Transparent Interconnection of Lots of Links (TRILL): Fine-Grained Labeling", RFC 7172, May 2014.

[RFC7176] Eastlake, D., Senevirathne, T., Ghanwani, A., Dutt, D., and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", RFC 7176, May 2014.

[RFC7357] Zhai, H., Hu, F., Perlman, R., Eastlake, D., and O. Stokes, "Transparent Interconnection of Lots of Links (TRILL): End Station Address Distribution Information (ESADI) Protocol", RFC 7357, September 2014.

Authors' Addresses

Radia Perlman
Intel Labs
2200 Mission College Blvd.
Santa Clara, CA 95054-1549
USA

Phone: +1-408-765-8080
Email: Radia@alum.mit.edu

Fangwei Hu
ZTE Corporation
No.889 Bibo Rd
Shanghai 201203
China

Phone: +86 21 68896273
Email: hu.fangwei@zte.com.cn

Donald Eastlake,3rd
Huawei technology
155 Beaver Street
Milford, MA 01757
USA

Phone: +1-508-634-2066
Email: d3e3e3@gmail.com

Kesava Vijaya Krupakaran
Dell
Olympia Technology Park
Guindy Chennai 600 032
India

Phone: +91 44 4220 8496
Email: Kesava_Vijaya_Krupak@Dell.com

Ting Liao
ZTE Corporation
No.50 Ruanjian Ave.
Nanjing, Jiangsu 210012
China

Phone: +86 25 88014227
Email: liao.ting@zte.com.cn