

TRILL working group
Internet Draft
Intended status: Standard Track
Expires: Sept 2014

L. Dunbar
D. Eastlake
Huawei
Radia Perlman
Intel
I. Gashinsky
Yahoo
July 15, 2013

Directory Assisted TRILL Encapsulation
draft-dunbar-trill-directory-assisted-encap-04.txt

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/ietf/lid-abstracts.txt>

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

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this

Internet-Draft Directory Assisted TRILL Encapsulation

document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

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

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 0.

The term ''TRILL'' and ''RBridge'' are used interchangeably in this document. The term ''subnet'' and ''VLAN'' are also used interchangeably because it is very common to map one subnet to one VLAN.

Table of Contents

1. Introduction	3
2. Terminology	3
3. Directory Assistance to Non-RBridge	3
4. Source Nickname in Frames Encapsulated by Non-RBridge Nodes..	6
5. Benefits of Non-RBridge encapsulating TRILL header	7
5.1. Avoid Nickname Exhaustion Issue	7
5.2. Reduce FDB size for switches on Bridged LANs	7
6. Conclusion and Recommendation	8
7. Manageability Considerations.....	8
8. Security Considerations.....	8
9. IANA Considerations	8
10. Acknowledgments	8
11. References	8
Authors' Addresses	9
Intellectual Property Statement.....	10
Disclaimer of Liability.....	10

Internet-Draft Directory Assisted TRILL Encapsulation

1. Introduction

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

[RBridge-directory] describes the framework for RBridge edge to get MAC&VLAN<->RBridgeEdge mapping from a directory service in data center environment instead of flooding unknown DAs across TRILL domain. When directory is used, any node, even non-RBridge node, can perform the TRILL encapsulation. This draft is to demonstrate the benefits of non-RBridge nodes performing TRILL encapsulation.

2. Terminology

AF Appointed Forwarder RBridge port

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

FDB: Filtering Database for Bridge or Layer 2 switch

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.

VM: Virtual Machines

3. Directory Assistance to Non-RBridge

With directory assistance [RBridge-Directory], a non-RBridge can determine if a packet needs to be forwarded across the RBridge domain. Suppose the RBridge domain boundary starts at

network switches (i.e. not virtual switches embedded on servers), a directory can assist Virtual Switches embedded on servers to encapsulate proper TRILL header by providing the information of the egress RBridge edge to which the target is attached. If a target is not attached to other RBridge edge nodes based on the directory [RBridge-Directory], the non-RBridge node can forward the data frames natively, i.e. not encapsulating any TRILL header.

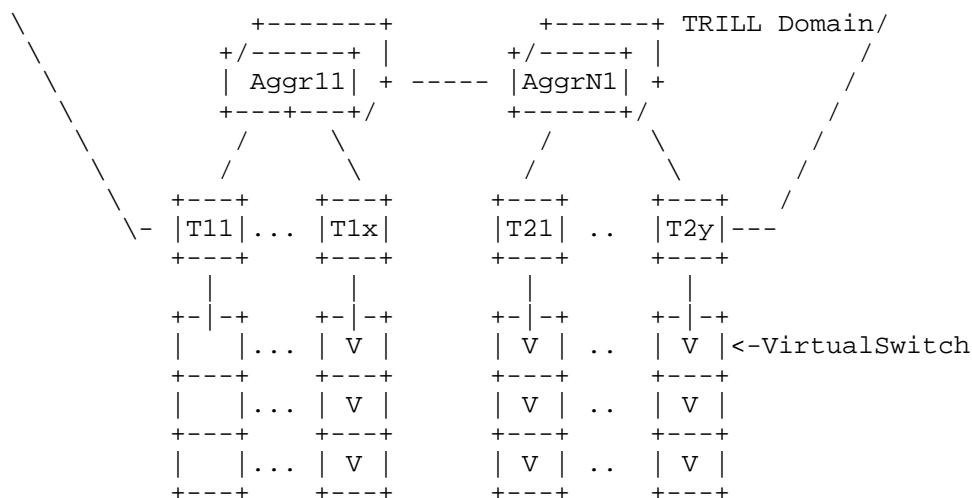


Figure 1: TRILL domain in typical Data Center Network

When a TRILL encapsulated data packet reaches the ingress RBridge, the ingress RBridge can simply forward the pre-encapsulated packet to the RBridge that is specified in the DA field of the TRILL header of the data frame. When the ingress RBridge receives a native Ethernet frame, it only forward the data frame to the directly attached bridged LAN.

Under this environment, the ingress RBridge doesn't flood or send the received Ethernet data frames to TRILL domain when the DA in the Ethernet data frames is unknown or instructed by the directory not to be sent across TRILL domain. Under this scheme, for an RBridge with multiple ports connected to a bridged LAN, data frames received from TRILL domain, decapsulated and forwarded to the bridged LAN via one port, and flooded back to the RBridge via another port, won't be encapsulated again and forwarded back TRILL domain.

Internet-Draft Directory Assisted TRILL Encapsulation

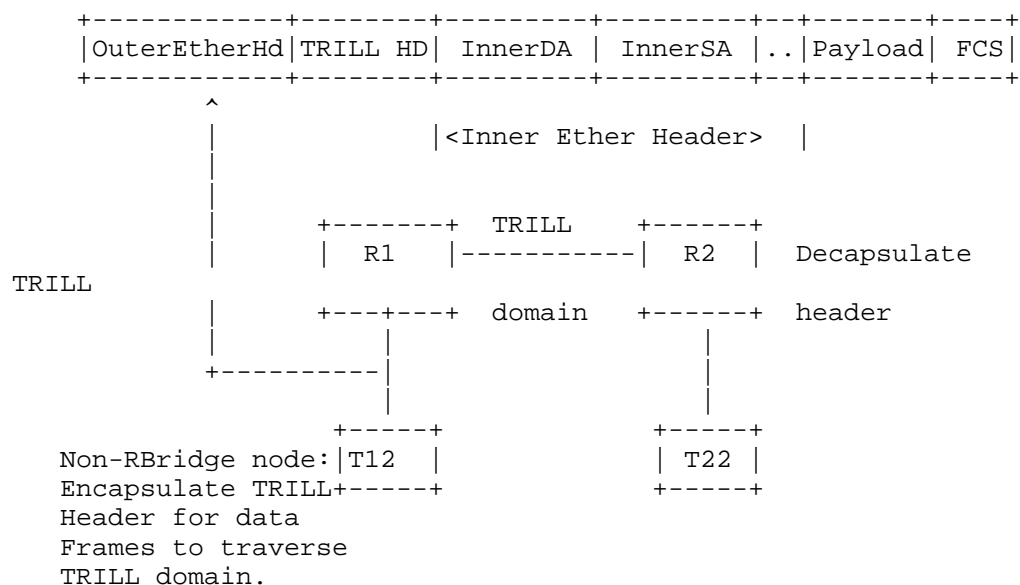
That means there is no need to worry about AF ports and all RBridge edge ports connected to one bridged LAN can receive and forward pre-encapsulated traffic, which greatly improves the overall network utilization.

Note: [RBridge] 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 data frames do not need to be sent across RBridge domain, they are switched by all nodes/ports per IEEE802.1Q and RBridge edge will not encapsulate and forward those data frames across RBridge domain.

When a pre-encapsulated TRILL frame arrives at an RBridge whose nickname matches with the destination nickname in the TRILL header, the processing is exactly same as normal, i.e. it decapsulates the received TRILL frame and forwards the decapsulated Ethernet frame to the target attached to its edge ports. If the DA of the decapsulated Ethernet frame is not in the egress RBridge's FDB, the egress RBridge can flood the decapsulated Ethernet frame to all hosts attached.

We call a node that only performs the TRILL encapsulation but doesn't participate in RBridge's IS-IS routing a "TRILL Encapsulating node" or "Simplified RBridge". The TRILL Encapsulating Node gets the MAC&VLAN<->RBridgeEdge mapping table pushed down or pulled from directory servers [RBridge-directory]. Upon receiving a native Ethernet frame, the TRILL Encapsulating Node 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 forwarded per IEEE802.1Q.



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 a non-RBridge node, using the Ingress RBridge edge node's nickname in the source address field will make the ingress RBridge node receive TRILL frames with its own nickname in the frames' source address field, which can be confusing.

To avoid confusion of edge R Bridges receiving TRILL encapsulated frames with their own nickname in the frames' source address field from neighboring non-R Bridge nodes, a new nickname can be given to an R Bridge edge node, e.g. Phantom Nickname, to represent all the TRILL Encapsulating Nodes attached to the R Bridge edge node.

When the Phantom Nickname is used in the Source Address field of a TRILL frame, it is understood that the TRILL encapsulation is actually done by a non-RBridge node which is attached to an edge port of an RBridge Ingress node.

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 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. 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 reasonable number of RBridge nodes in a large data center. All the TRILL encapsulating nodes attached to one RBridge are represented by one TRILL nickname, i.e. Phantom Nickname, which avoids the Nickname exhaustion problem.

5.2. Reduce FDB size 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 $200 \times 4000 = 800,000$ entries.

However, if the virtual switches on server pre-encapsulate the data frames towards hosts attached to other RBridge Edge nodes with TRILL header, 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 RBridge edges.

Internet-Draft Directory Assisted TRILL Encapsulation

There are multiple ways for local switches to avoid adding remote hosts' MAC to their FDB. One simple way is by disabling learning on source addresses. The local switches can be pre-installed with MAC addresses of local hosts with the assistance of directory.

6. Conclusion and Recommendation

When directory service is available, nodes outside TRILL domain become capable of encapsulating TRILL header for data frames destined for remote RBridges that is not on the same bridged LAN. 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

TBD.

8. Security Considerations

TBD.

9. IANA Considerations

TBD

10. Acknowledgments

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

11. References

[RBridge-Directory] Dunbar, et, al ''TRILL (Transparent Interconnection of Lots of Links) Edge Directory Assistance Framework'', < draft-ietf-trill-directory-framework-03>, March, 2013

Internet-Draft Directory Assisted TRILL Encapsulation

[RBridges] Perlman, et, al ''RBridge: Base Protocol Specification'', <draft-ietf-trill-rbridge-protocol-16.txt>, March, 2010

[RBridges-AF] Perlman, et, al ''RBridges: Appointed Forwarders'', <draft-ietf-trill-rbridge-af-02.txt>, April 2011

[ARMD-Problem] Dunbar, et,al, ''Address Resolution for Large Data Center Problem Statement'', Oct 2010.

[ARP reduction] Shah, et. al., "ARP Broadcast Reduction for Large Data Centers", Oct 2010

Authors' Addresses

Linda Dunbar
Huawei Technologies
1700 Alma Drive, Suite 500
Plano, TX 75075, USA
Phone: (972) 543 5849
Email: ldunbar@huawei.com

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

Internet-Draft Directory Assisted TRILL Encapsulation

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

Intellectual Property Statement

The IETF Trust takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in any IETF Document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.

Copies of Intellectual Property disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement any standard or specification contained in an IETF Document. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Liability

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE

Internet-Draft Directory Assisted TRILL Encapsulation

INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

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

TRILL

Weiguo Hao
Yizhou Li
Donald Eastlake
Huawei
February 14, 2014

Internet Draft
Intended status: Informational
Expires: August 2014

Analysis of Active-Active connection solutions
draft-hao-trill-analysis-active-active-01.txt

Status of this Memo

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

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/ietf/lid-abstracts.txt>

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

This Internet-Draft will expire on October 14, 2014.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

Draft [TRILL-Active-PS] lists basic problems which any active-active solutions should address, these problems include frame duplications, loop, MAC address flip-flop and unsynchronized information among member R Bridges. For each problem, there may be multiple ways to deal with it. Some solutions solve all or most of the problems listed, and at the same time introduces extra issues. This draft tries to analyze and compare the different solutions for each of the issues, gives a brief summary on the pros and cons, and/or the applicable scenarios.

Table of Contents

1. Introduction	3
2. Conventions used in this document.....	5
3. Frame duplications	5
4. Loop.....	6
4.1. Independent nickname allocation.....	6
4.2. Consistent nickname allocation.....	6
4.3. Comparison	7
5. Address flip-flop	7
5.1. Data plane learning mode.....	7
5.1.1. CMT	8
5.1.2. Centralized replication.....	8
5.1.3. Tunneling among edge RBs.....	9
5.1.4. Comparison.....	9
5.2. Control plane learning mode	10
6. Unsynchronized information among member RBridges	10
6.1. RBridge channel based communication protocol	11
6.2. TRILL LSP extension	11
6.3. Comparison	11
7. Solution summary	11
8. Security Considerations	13
9. IANA Considerations	13
10. References	13
10.1. Normative References	13
10.2. Informative References	13

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.

Customer edge(CE) devices typically are multi-homed to several RBridges. All of the uplinks of a CE are considered as an Multi-Chassis Link Aggregation (MC-LAG) bundle. An edge group is the group of edge RBridges that a CE is multi-homed to in active-active mode. An edge group corresponds to an MC-LAG. One RB can be in more than one edge group. An active-active flow-based load-sharing mechanism is desirable to achieve better load balancing and high reliability. A CE device can be a layer3 end system by itself or a bridge switch through which layer3 end systems are accessed to TRILL campus.

Draft [TRILL-Active-PS] lists the following problems which any active-active solution should address:

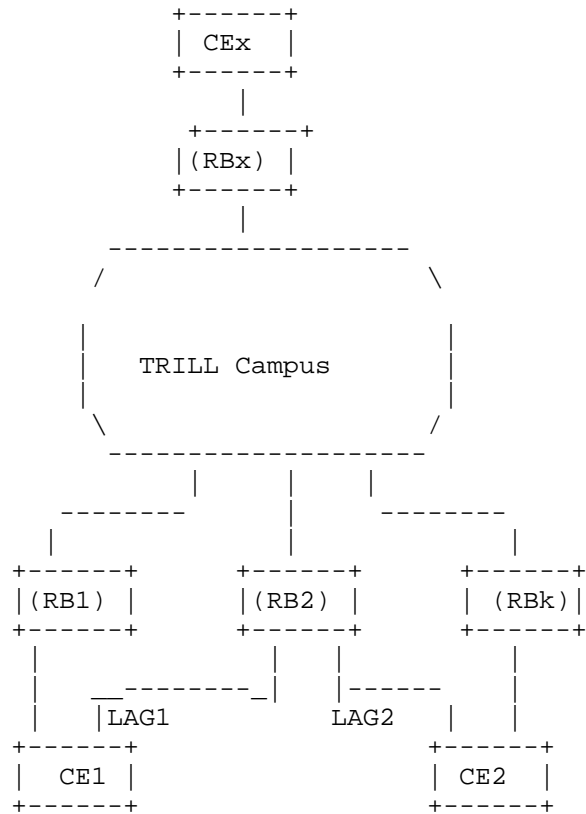


Figure 1 TRILL Active-Active Access Scenario

1. Frame duplications
2. Loop
3. Address flip-flop
4. Unsynchronized information among member RBridges

For each problem, there may be multiple ways to deal with it. And some solutions solve all or most of the problems listed, and at the same time introduces extra issues. This draft tries to analyze and compare the different solutions for each of the issue, gives a brief

summary on the pros and cons, and/or the applicable scenarios. The co-authors believe such analysis is helpful to design a more completed solution in future.

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 - Customer equipment. Could be a bridge or end station or a hypervisor.

CMT - Coordinated Multicast Trees [CMT].

Edge group - a group of edge RBs to which at least one CE is multiply attached. One RB can be in more than one edge group.

LACP - Link Aggregation Control Protocol.

LAG - Link Aggregation, as specified in [8021AX].

3. Frame duplications

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. DF 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.

Each RB in an edge group elects a DF using same algorithm which guarantees the same RB elected as DF per MC-LAG per VLAN. The RB that is elected as a DF for a given VLAN will forward multi-destination traffic in the egress direction towards the CE. All non-DF RBs drop multi-destination traffic in the egress direction towards the CE. All edge RBs, including DF and non-DF, can ingress

the traffic to TRILL campus as usual.[draft-hao-trill-dup-avoidance-active-active-00] describes the detail DF mechanism and TRILL protocol extension for DF election.

4. Loop

If a CE sends a broadcast, unknown unicast, or multicast (BUM) packet to DF RB, it will forward that packet to all or subset of the other RBs including the non-DF RBs. Because non-DF RBs don't egress BUM frame to local access side, in this case the frame won't loop back to the CE.

If a CE sends a BUM packet to one of the non-DF (Designated Forwarder) RBs, say RB1, then RB1 will forward that packet to all or subset of the other RBs including the DF RB 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.

4.1. Independent nickname allocation

Each ingress RBridge allocates a unique nickname for each MC-LAG independently. It is not required that the nickname provisioned on all involved edge RBridges remains the same for one corresponding MC-LAG.

When the ingress RBridge receives a BUM frame from a local CE, it uses the nickname as ingress nickname for TRILL tunnel encapsulation and sends the frame to other RBridge(s).

When an egress RBridge receives a multicast frame from the TRILL campus, it checks the ingress nickname in the TRILL header and filters out the frame on all local interfaces connected to the same CE. Each egress RBridge should track the nickname(s) associated with the other RBridge(s) with which it has a shared multi-homed LAG. The solution has limited nickname allocation scalability issue, because each RBridge needs allocate per nickname per MC-LAG.

4.2. Consistent nickname allocation

Edge RBridges forming an MC-LAG in an edge group are assigned a globally unique pseudo-nickname. If multiple MC-LAGs exist, edge RBridges for each individual MC-LAG should be assigned such a

pseudo-nickname. It should be guaranteed that pseudo-nickname provisioned on all involving edge RBridges remains the same for one corresponding MC-LAG.

When a ingress RBridge receives traffic from a active-active accessed CE, it performs TRILL encapsulation with the pseudo-nickname as ingress nickname. When the traffic comes to each egress RBridge, the egress RBridge checks the ingress nickname in TRILL header and filters out the frame on all local interfaces connected to the same CE. Each egress RBridge relies on the pseudo-nickname to filter out the frame on all local interfaces connected to the same CE.

4.3. Comparison

	Solution	Independent Allocation	Consistent
Allocation			
1	Nickname consumption	High	Normal
	Scalability	Low	High

5. Address flip-flop

MAC learning in TRILL can be performed either in data plane or control plane. When a local host h1 attaches to multiple edge RBridges, learning at the remote host for h1 may have MAC flip-flop problem. There are different ways to avoid this for data plane learning and control plane learning scenarios.

5.1. Data plane learning mode

For data plane learning mode, to avoid mac address flip-flop on remote RBs, a pseudo-nickname [TRILLPN] solution was proposed. The basic idea is to represent all member links of the MC-LAG as a virtual RBridge with single pseudo-nickname. Any member RBridge of the MC-LAG should use this pseudo-nickname rather than its own nickname as ingress nickname when inject TRILL data frames. It solves the above mentioned problems pretty well; however, it

introduces another issue: packet drop due to RPF check. To overcome the RPF check failure issue, three solutions have been proposed.

5.1.1.1. CMT

CMT [CMT] solution allows edge RBridges to specify different distribution trees to forward BUM traffic from a connecting CE device by using a new IS-IS Affinity sub-TLV. Remote RBridges calculate their forwarding tables and derive the RPF for distribution trees based on the distribution tree association advertisements.

In this solution, it's required to establish multiple distribution trees in a TRILL campus, i.e. if a CE is active-active accessed to 4 edge RBridges, at least 4 distribution trees are required. No hardware upgrade is needed for RBridges in the TRILL campus, only software upgrade is needed.

5.1.2. Centralized replication

Ingress RB participating in active-active connection sends BUM traffic to one of a distribution tree root node through unicast TRILL encapsulation. The distribution tree root node acts as centralized replication node. When the distribution tree root node receives unicast TRILL encapsulation BUM traffic from the ingress RB, it decapsulates the unicast TRILL packet. Then it replicates and forwards the BUM traffic to all other destination RBs through the distribution tree established per TRILL base protocol. [draft-hao-trill-centralized-replication-00] describes the detail centralized replication solution. Through the centralized replication solution, only unicast forwarding behavior is required between edge RB and distribution tree root RB, so no RPF check function is required along the path between ingress RB and distribution tree node.

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. Then it is replicated and forwarded to other CE devices through TRILL distribution tree, even when the receiver CE is connected to the same RBridge as the sender CE. To avoid duplicated traffic on receiver CE, ingress RBridge can't locally replicate and forward the BUM traffic to other connecting CE when it receives BUM traffic from an active-active sender CE, i.e. the access port of the ingress RBridge should be isolated from other local access ports.

In this solution, it's required to consume more network bandwidth between ingress RB and distribution tree root node than CMT solution.

Both hardware and software upgrade are required on edge RBs participating in active-active connection and the distribution tree root node. This solution doesn't require multiple distribution trees in TRILL campus, so it has better scalability than CMT.

5.1.3. Tunneling among edge RBs

This solution allows only a selected edge RBridge in an edge group participating in active-active access to be responsible for forwarding BUM traffic from connecting CE to TRILL campus along distribution tree per TRILL base protocol. All other edge RBridges in the virtual RBridge send BUM traffic from connecting CE to the selected edge RBridge through unicast TRILL encapsulation. When the selected edge RBridge receives TRILL traffic from other RBs in a same virtual RBridge, the selected RB decapsulates the unicast TRILL packet. Then it forwards the BUM traffic to trill campus along distribution tree established per TRILL protocol.

Similar to the solution of centralized replication, to avoid duplicated traffic on receiver CE, the access port of ingress RBridge connecting to an active-active accessing sender CE should be isolated from other local access ports.

In this solution, it's required to consume more network bandwidth among edge RBs. Both hardware and software upgrade are required on edge RBs participating active-active connection. This solution doesn't require multiple distribution trees in TRILL campus, so it has better scalability than CMT.

5.1.4. Comparison

Solution		CMT	Centralized replication	Tunneling among edge RBs
Scalability		Medium	High	High
Network bandwidth consumption		Low	High	High

6.1. RBridge channel based communication protocol

RBridge channel based communication protocol among all RBridges in a edge group is introduced to implement synchronization. The communication protocol is restricted to RBridge nodes in each edge group, other RBridges in TRILL campus needn't involve. A new type of RBridge Channel message should be given by a Protocol field in the RBridge Channel Header to indicate synchronization information in the payload. RBridge channel message is forwarded through TRILL data plane. Transmission delay is relatively low.

6.2. TRILL LSP extension

TRILL LSP can be extended to implement synchronization among all edge RBridges. Synchronization information is conveyed through new TLVs or sub-TLVs in TRILL LSP. Because TRILL LSP is flooded to all RBridges in TRILL campus, so it may cause campus wide fluctuation. TRILL LSP is forwarded through control plane. Transmission delay is relatively high.

6.3. Comparison

+-----+-----+-----+			
-----+			
	Solution	RBridge channel based	TRILL LSP e
xtension			
+-----+-----+-----+			
-----+			
	Flooding scope	Edge group	Campus w
ide			
+-----+-----+-----+			
-----+			
	Forwarding	Data plane	Control p
lane			
+-----+-----+-----+			
-----+			

7. Solution summary

Through the above analysis, a completed solution for active-active connection can be stitched together using mechanisms for each individual problem analyzed in this draft.

If there are multiple mechanisms for a single problem, any one can be picked up. For example, in MAC learning through data plane scenarios for address flip-flop problem, there are three mechanisms including CMT, centralized replication and tunneling among edge RBs to solve MAC address flip-flop problems. Any one out of three can be

selected to combine with other mechanisms to form a whole solution. If there is only one mechanism for a single problem, then it is a mandatory part of the completed solution. For example, DF election mechanism is the only acceptable way to prevent frame duplication. Thus it is a mandatory part of the completed solution.

In summary, the whole solution for TRILL active-active connection is as follows.

Problem		Solution	
Data plane	Frame duplication	DF election	
	Loop	Data plane MAC learning	Control plane
Control plane	MAC learning	MAC learning	MAC learning
		CMT Centralized Tunneling	
Control plane		replication	among edge RBs
	Address flip-flop allocation	Independent allocation	Consistent allocation
Control plane	Unsynchronized information	RBridge channel based	LSP extension

8. Security Considerations

This draft does not introduce any extra security risks. 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

- [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.

10.2. Informative References

- [4] [TRILAA] Li, Y., et.al., "Problems of Active-Active connection at the TRILL Edge", draft-yizhou-trill-active-active-connection-prob2, Work in progress, July 2013.
- [5] [TRILLPN] Zhai, H., et.al., "RBridge: Pseudonode Nickname", draft-hu-trill-pseudonode-nickname, Work in progress, November 2011.
- [6] [CMT] Senevirathne, T., Pathangi, J., and J. Hudson, "Coordinated Multicast Trees (CMT) for TRILL", draft-ietf-trill-cmt-01.txt Work in Progress, November 2012
- [7] [RFCchannel] - D. Eastlake, V. Manral, L. Yizhou, S. Aldrin, D. Ward, "TRILL: RBridge Channel Support", draft-ietf-trill-rbridge-channel-08.txt, in RFC Editor's queue.
- [8] [RFC6439] Perlman, R., Eastlake, D., Li, Y., Banerjee, A., and F. Hu, "Routing Bridges (RBridges): Appointed Forwarders", RFC 6439, November 2011.

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

Donald Eastlake 3rd
Huawei Technologies
155 Beaver Street
Milford, MA 01757 USA
Phone: +1-508-333-2270
EMail: d3e3e3@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/lid-abstracts.html>. The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Table of Contents

1. Introduction.....	3
1.1 Terminology.....	3
2. Push Model Directory Assistance Mechanisms.....	5
2.1 Requesting Push Service.....	5
2.2 Push Directory Servers.....	5
2.3 Push Directory Server State Machine.....	6
2.3.1 Push Directory States.....	6
2.3.2 Push Directory Events and Conditions.....	7
2.3.3 State Transition Diagram and Table.....	8
2.4 Additional Push Details.....	9
2.5 Primary to Secondary Server Push Service.....	10
3. Pull Model Directory Assistance Mechanisms.....	12
3.1 Pull Directory Message Common Format.....	12
3.2 Pull Directory Query and Response Messages.....	14
3.2.1 Pull Directory Query Message Format.....	14
3.2.2 Pull Directory Response Format.....	16
3.3 Cache Consistency.....	19
3.3.1 Update Message Format.....	21
3.3.2 Acknowledge Message Format.....	22
3.4 Pull Directory Hosted on an End Station.....	22
3.5 Pull Directory Message Errors.....	23
3.6 Additional Pull Details.....	25
4. Events That May Cause Directory Use.....	26
4.1 Forged Native Frame Ingress.....	26
4.2 Unknown Destination MAC.....	26
4.3 Address Resolution Protocol (ARP).....	27
4.4 IPv6 Neighbor Discovery (ND).....	28
4.5 Reverse Address Resolution Protocol (RARP).....	28
5. Layer 3 Address Learning.....	29
6. Directory Use Strategies and Push-Pull Hybrids.....	30
6.1 Strategy Configuration.....	30
7. Security Considerations.....	33
8. IANA Considerations.....	34
8.1 ESADI-Parameter Data Extensions.....	34
8.2 RBridge Channel Protocol Number.....	35
8.3 The Pull Directory (PUL) and No Data (NOD) Bits.....	35
Acknowledgments.....	36
Normative References.....	37
Informational References.....	38
Authors' Addresses.....	39

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:

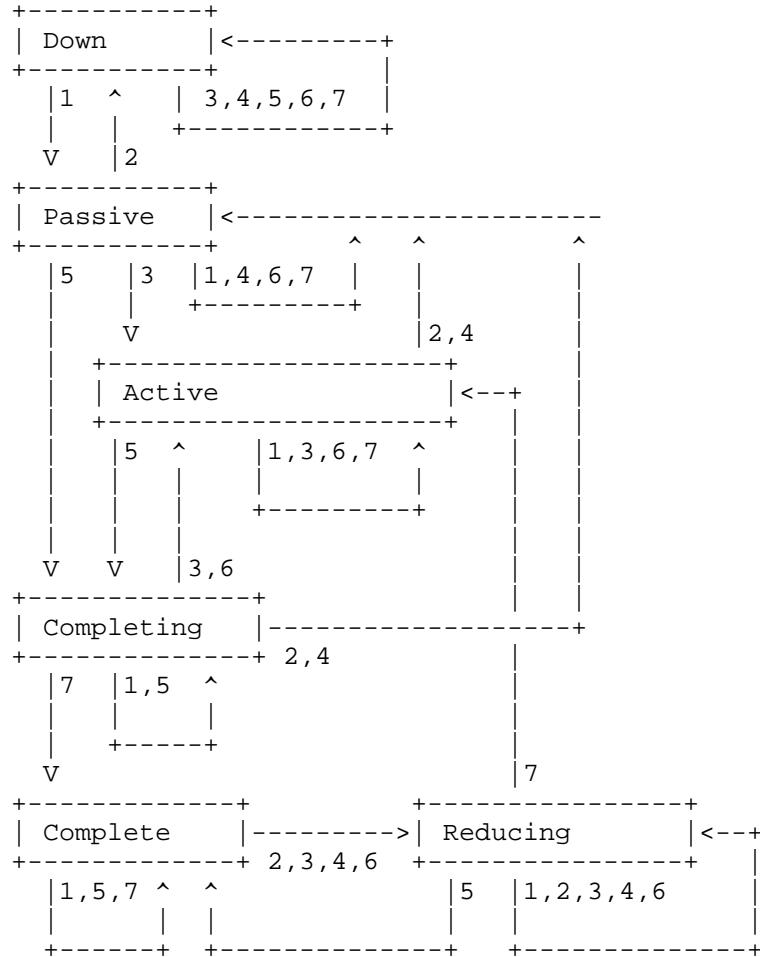


Figure 1. Push Server State Diagram

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 multideestination 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 send 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:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Ver  | Type | Flags | Count |      Err      |      SubErr      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Sequence Number                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Type Specific Payload - variable length
+---+---+ ...

```

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:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Ver  | Type | Flags | Count |      Err      |      SubErr      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Sequence Number                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
| QUERY 1
+-----+-----+-----+-----+-----+-----+-----+...
| QUERY 2
+-----+-----+-----+-----+-----+-----+-----+...
| ...
+-----+-----+-----+-----+-----+-----+-----+...
| QUERY K
+-----+-----+-----+-----+-----+-----+-----+...

```

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:

```

      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           SIZE           |   RESV   |   QTYPE   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
If QTYPE = 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           AFN           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Query address ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
If QTYPE = 2, 3, 4, or 5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Query frame ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

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 send 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:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Ver   | Type  | Flags | Count |      Err      |      SubErr    |
+-----+-----+-----+-----+-----+-----+-----+
|                               Sequence Number                               |
+-----+-----+-----+-----+-----+-----+-----+
| RESPONSE 1
+-----+-----+-----+-----+-----+-----+...
| RESPONSE 2
+-----+-----+-----+-----+-----+-----+...
| ...
+-----+-----+-----+-----+-----+-----+...
| RESPONSE K
+-----+-----+-----+-----+-----+-----+...

```

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:

```

      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
+-----+-----+-----+-----+-----+-----+-----+-----+
|          SIZE          |OV|  RESV  |   Index   |
+-----+-----+-----+-----+-----+-----+-----+
|                               Lifetime                               |
+-----+-----+-----+-----+-----+-----+-----+
|                               Response Data ...                               |
+-----+-----+-----+-----+-----+-----+-----+...

```

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 | P | N | R |
+---+---+---+---+
```

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 Channel 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:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Ver   | Type  | Flags | Count | Err     | SubErr   |
+-----+-----+-----+-----+-----+-----+-----+
|                               Sequence Number                               |
+-----+-----+-----+-----+-----+-----+-----+
| Nickname (2 bytes) |
+-----+-----+-----+-----+-----+-----+-----+...+
| Data Label ... (4 or 8 bytes) |
+-----+-----+-----+-----+-----+-----+-----+...+
| Type Specific Payload - variable length |
+-----+ ...

```

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 receive 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 than 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 than 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 hopes 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:

```

+-----+
| Type | (1 byte)
+-----+
| Length | (1 byte)
+-----+
|R| Priority | (1 byte)
+-----+
| CSNP Time | (1 byte)
+-----+
| Flags | (1 byte)
+-----+
|PushDirPriority| (optional, 1 byte)
+-----+
| Reserved for expansion | (variable)
+-----+
+-----+

```

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

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License. The definitive version of an IETF Document is that published by, or under the auspices of, the IETF. Versions of IETF Documents that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of IETF Documents. The definitive version of these Legal Provisions is that published by, or under the auspices of, the IETF. Versions of these Legal Provisions that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of these Legal Provisions. For the avoidance of doubt, each Contributor to the IETF Standards Process licenses each Contribution that he or she makes as part of the IETF Standards Process to the IETF Trust pursuant to the provisions of RFC 5378. No language to the contrary, or terms, conditions or rights that differ from or are inconsistent with the rights and licenses granted under RFC 5378, shall have any effect and shall be null and void, whether published or posted by such Contributor, or included with or in such Contribution.

INTERNET-DRAFT
Intended status: Proposed Standard

Donald Eastlake
Yizhou Li
Huawei
Radia Perlman
Intel
December 13, 2013

Expires: June 12, 2014

TRILL: Interface Addresses APPsub-TLV
<draft-ietf-trill-ia-appsubtlv-00.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.

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

Table of Contents

1. Introduction.....	3
1.1 Conventions Used in This Document.....	3
2. Format of the Interface Addresses APPsub-TLV.....	5
3. IA APPsub-TLV sub-sub-TLVs.....	10
3.1 AFN Size sub-sub-TLV.....	10
3.2 Fixed Address sub-sub-TLV.....	11
3.3 Data Label sub-sub-TLV.....	12
3.4 Topology sub-sub-TLV.....	12
4. Security Considerations.....	14
5. IANA Considerations.....	15
5.1 Additional AFN Number Allocation.....	15
5.2 IA APPsub-TLV Sub-Sub-TLVs SubRegistry.....	16
Acknowledgments.....	17
Appendix A: Examples.....	18
A.1 Simple Example.....	18
A.2 Complex Example.....	18
Normative References.....	21
Informational References.....	22
Authors' Addresses.....	23

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 [RFCfgl])) 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 Considerations 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 [RFCfgl]

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.

```

+-----+-----+
| Type = TBD                                     | (2 bytes)
+-----+-----+
| Length                                         | (2 bytes)
+-----+-----+
| Addr Sets End                                 | (2 bytes)
+-----+-----+
| Nickname                                       | (2 bytes)
+-----+-----+
| Flags                                         | (1 byte)
+-----+-----+
| Confidence                                    | (1 byte)
+-----+-----+
| Template ...                                 (variable)
+-----+-----+...+
| Address Set 1 (size determined by Template) |
+-----+-----+...+
| Address Set 2 (size determined by Template) |
+-----+-----+...+
| ...
+-----+-----+...+
| Address Set N (size determined by Template) |
+-----+-----+...+
| optional sub-sub-TLVs ...
+-----+-----+...

```

Figure 1. The Interface Addresses APPsub-TLV

- o Type: Interface Addresses TRILL APPsub-TLV type, set to TBD[#2 suggested] (IA-SUBTLV).
- o 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.
- o 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.
- o 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).
- o Flags: A byte of flags as follows:

```

  0 1 2 3 4 5 6 7
+-----+
|D|L|N|  RESV  |
+-----+

```

D: Directory flag: If D is one, the APPsub-TLV contains Directory information [RFC7067].

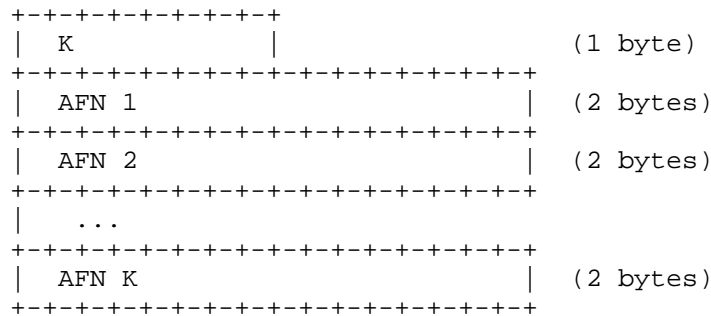
L: Local flag: If L is one, the APPsub-TLV contains information learned locally by observing ingressed frames [RFC6325]. (Both D and L can one in the same IA APPsub-TLV if a TRILL switch that had learned an address locally also advertised it as a directory.)

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.

- o 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.
- o 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 field 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.

- o 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.
- o 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.

```

+++++
| Type = AFNsz                               | (2 byte)
+++++
| Length                                     | (2 byte)
+++++
| AFN Size Record 1                         | (3 bytes)
+++++
| AFN Size Record 2                         | (3 bytes)
+++++
| ...
+++++
| AFN Size Record N                         | (3 bytes)
+++++

```

Where each AFN Size Record is structured as follows:

```

+-----+
|  AFN                                     | (2 bytes)
+-----+
|  AddrSize                               | (1 byte)
+-----+

```

- o Type: AFN-Size sub-sub-TLV type, set to 1 (AFNsz).
- o 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.
- o AFN Size Record(s): Zero or more 3-byte records, each giving the size of an address type identified by an AFN,
- o AFN: The AFN whose length is being specified by the AFN Size Record.
- o AddrSize: 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=FIXEDADR                           | (2 byte)
+-----+
| Length                                  | (2 byte)
+-----+
| AFN                                     | (2 bytes)
+-----+
| Fixed Address                           | (variable)
+-----+

```

- o Type: Data Label sub-sub-TLV type, set to 2 (FIXEDADR).
- o Length: variable, minimum 3. If Length is 2 or less, the sub-sub-

TLV MUST be ignored.

- o AFN: Address Family Number of the Fixed Address.
- o 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=DATALEN                               | (2 byte)
+-----+
| Length                                   | (2 byte)
+-----+
| Data Label                               | (variable)
+-----+

```

- o Type: Data Label sub-TLV type, set to 3 (LABEL).
- o Length: 2 or 3. If Length is some other value, the sub-sub-TLV MUST be ignored.
- o 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 [RFCfgl].

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=DATALEN| (2 byte)
+-----+
| Length| (2 byte)
+-----+
| RESV | Topology | (2 bytes)
+-----+

```

- o Type: Topology sub-TLV type, set to 4 (TOPOLOGY).
- o 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.

- o 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 [RFCfgl]) 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 [RFCchannel], 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-bit MAC 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-bit MAC 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 0xFFFFE 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

0x20010DB800000000000005EFFF0053DE IPv6 Address

Address Set Two

0x20010DB800000000000005EFFF0053E3 IPv6 Address

Address Set Three

0x20010DB800000000000005EFFF0053D3 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 then be supplied by a second Fixed Address sub-sub-TLV providing 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.

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.
- [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.
- [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.
- [RFCchannel] - D. Eastlake, V. Manral, Y. Li, S. Aldrin, D. Ward, "TRILL: RBridge Channel Support", draft-ietf-trill-rbridge-channel, 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.

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

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License. The definitive version of an IETF Document is that published by, or under the auspices of, the IETF. Versions of IETF Documents that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of IETF Documents. The definitive version of these Legal Provisions is that published by, or under the auspices of, the IETF. Versions of these Legal Provisions that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of these Legal Provisions. For the avoidance of doubt, each Contributor to the IETF Standards Process licenses each Contribution that he or she makes as part of the IETF Standards Process to the IETF Trust pursuant to the provisions of RFC 5378. No language to the contrary, or terms, conditions or rights that differ from or are inconsistent with the rights and licenses granted under RFC 5378, shall have any effect and shall be null and void, whether published or posted by such Contributor, or included with or in such Contribution.

TRILL Working Group
Internet Draft
Intended status: Standards Track
Expires: August 2014

T. Mizrahi
Marvell
T. Senevirathne
S. Salam
D. Kumar
Cisco
D. Eastlake 3rd
Huawei
February 11, 2014

Loss and Delay Measurement in
Transparent Interconnection of Lots of Links (TRILL)
<draft-ietf-trill-loss-delay-02.txt>

Abstract

Performance Monitoring (PM) is a key aspect of Operations, Administration and Maintenance (OAM). It allows network operators to verify the Service Level Agreement (SLA) provided to customers, and to detect network anomalies. This document specifies mechanisms for Loss Measurement and Delay Measurement in TRILL networks.

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/ietf/lid-abstracts.txt>.

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

This Internet-Draft will expire on August 11, 2014.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in this Document	4
2.1. Keywords	4
2.2. Definitions	4
2.3. Abbreviations	5
3. Loss and Delay Measurement in the TRILL Architecture	6
3.1. Performance Monitoring Granularity	6
3.2. One-Way vs. Two-Way Performance Monitoring	7
3.2.1. One-Way Performance Monitoring	7
3.2.2. Two-Way Performance Monitoring	7
3.3. Point-to-point vs. Point-to-multipoint PM	8
4. Loss Measurement	8
4.1. One-Way Loss Measurement	8
4.1.1. 1SL Message Transmission	9
4.1.2. 1SL Message Reception	10
4.2. Two-Way Loss Measurement	11
4.2.1. SLM Message Transmission	12
4.2.2. SLM Message Reception	12
4.2.3. SLR Message Reception	13
5. Delay Measurement	14
5.1. One-Way Delay Measurement	14
5.1.1. 1DM Message Transmission	15
5.1.2. 1DM Message Reception	16
5.2. Two-Way Delay Measurement	16
5.2.1. DMM Message Transmission	17
5.2.2. DMM Message Reception	17
5.2.3. DMR Message Reception	18
6. Packet Formats	19
6.1. TRILL OAM Encapsulation	19
6.2. Loss Measurement Packet Formats	21

6.2.1. Counter Format	21
6.2.2. 1SL Packet Format	22
6.2.3. SLM Packet Format	23
6.2.4. SLR Packet Format	24
6.3. Delay Measurement Packet Formats	25
6.3.1. Timestamp Format	25
6.3.2. 1DM Packet Format	25
6.3.3. DMM Packet Format	26
6.3.4. DMR Packet Format	27
6.4. OpCode Values	28
7. Performance Monitoring Process	28
8. Security Considerations	29
9. IANA Considerations	29
10. Acknowledgments	29
11. References	30
11.1. Normative References	30
11.2. Informative References	30

1. Introduction

TRILL [RFCTRILL] is a protocol for transparent least cost routing, where RBridges route traffic to their destination based on least cost, using a TRILL encapsulation header with a hop count.

Operations, Administration and Maintenance (OAM) [OAM] is a set of tools for detecting, isolating and reporting connection failures and performance degradation. Performance Monitoring (PM) is a key aspect of OAM. PM allows network operators to detect and debug network anomalies and incorrect behavior. PM consists of two main building blocks - Loss Measurement and Delay Measurement. PM may also include other derived metrics such as Packet Delivery Rate, and Inter-Frame Delay Variation.

The requirements of OAM in TRILL networks are defined in [OAM-REQ], and the TRILL OAM framework is described in [OAM-FRAMEWK]. These two documents also highlight the main requirements in terms of performance monitoring.

This document defines protocols for loss measurement and for delay measurement in TRILL networks. These protocols generally conform to the performance monitoring functionality defined in ITU-T G.8013/Y.1731 [Y.1731].

- o Loss Measurement: the Loss Measurement protocol measures packet loss between two RBridges. The measurement is performed by sending a set of synthetic packets, and counting the number of packets transmitted and received during the test. The frame loss is calculated by comparing the numbers of transmitted and received packets. This provides a statistical estimate of the packet loss between the involved RBridges, with a margin of error that can be controlled by varying the number of transmitted synthetic packets. This document does not define procedures for packet loss computation based on counting user data. For further details see [OAM-FRAMEWK].
- o Delay Measurement: the Delay Measurement protocol measures the packet delay and packet delay variation between two RBridges. The measurement is performed using timestamped OAM messages.

2. Conventions Used in this Document

2.1. Keywords

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 [KEYWORDS].

The requirement level of PM in [OAM-REQ] is 'SHOULD'. Nevertheless, this memo uses the entire range of requirement levels, including 'MUST'; the requirements in this memo are to be read as 'A MEP (Maintenance End Point) that implements TRILL PM MUST/SHOULD/MAY/...'.

2.2. Definitions

- o One-way packet delay - (based on [IPPM-1DM]) the time elapsed from the start of transmission of the first bit of a packet by an RBridge until the reception of the last bit of the packet by the remote RBridge.
- o Two-way packet delay - (based on [IPPM-2DM]) the time elapsed from the start of transmission of the first bit of a packet from the local RBridge, receipt of the packet at the remote RBridge, the remote RBridge sending a response packet back to the local RBridge and the local RBridge receiving the last bit of that response packet.

- o Packet loss - (based on [IPPM-Loss]) the number of packets sent by a source RBridge and not received by the destination Rbridge. In the context of this document, packet loss is measured at a specific probe instance, and a specific observation period. As in [Y.1731], this document distinguishes between near-end and far-end packet loss. Note that this semantic distinction specifies the direction of packet loss, but does not affect the nature of the packet loss metric, which is defined in [IPPM-Loss].
- o Far-end packet loss - the number of packets lost on the path from the local RBridge to the remote RBridge in a specific probe instance, and a specific observation period.
- o Near-end packet loss - the number of packets lost on the path from the remote RBridge to the local RBridge in a specific probe instance, and a specific observation period.

2.3. Abbreviations

1DM	One-way Delay Measurement message
1SL	One-way Synthetic Loss Measurement message
DMM	Delay Measurement Message
DMR	Delay Measurement Reply
FGL	Fine Grained Label [RFC-FGL]
MD	Maintenance Domain
MD-L	Maintenance Domain Level
MEP	Maintenance End Point
MIP	Maintenance Intermediate Point
MP	Maintenance Point
OAM	Operations, Administration and Maintenance
PM	Performance Monitoring
SLM	Synthetic Loss Measurement Message
SLR	Synthetic Loss Measurement Reply

TLV Type, Length and Value

TRILL Transparent Interconnection of Lots of Links [RFC6713]

3. Loss and Delay Measurement in the TRILL Architecture

As described in [OAM-FRAMEWK], OAM protocols in a TRILL campus operate over two types of Maintenance Points (MPs): Maintenance End Points (MEPs) and Maintenance Intermediate Points (MIPs).

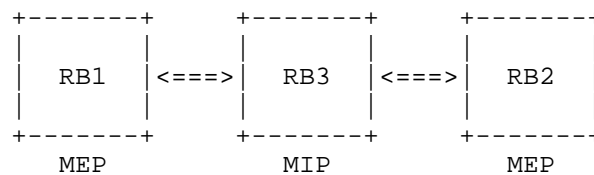


Figure 1 Maintenance Points in a TRILL Campus

Performance Monitoring (PM) allows a MEP to perform loss and delay measurements to any other MEP in the campus. Performance Monitoring is performed in the context of a specific Maintenance Domain (MD).

The PM functionality defined in this document is not applicable to MIPs.

3.1. Performance Monitoring Granularity

As defined in [OAM-FRAMEWK], PM can be applied at three levels of granularity: 'Network', 'Service' and 'Flow'.

- o Network-level PM: the PM protocol is run over a dedicated test VLAN or FGL.
- o Service-level PM: the PM protocol is used to perform measurements of actual user VLANs or FGL.
- o Flow-level PM: the PM protocol is used to perform measurements on a per-flow basis. A flow, as defined in [OAM-REQ], is a set of packets that share the same path and per-hop behavior (such as priority). As defined in [OAM-FRAMEWK], flow-based monitoring uses a Flow Entropy field that resides at the beginning of the OAM packet header (see Section 6.1.), and mimics the forwarding behavior of the monitored flow.

3.2. One-Way vs. Two-Way Performance Monitoring

Paths in a TRILL network are not necessarily symmetric, that is, a packet sent from RB1 to RB2 does not necessarily traverse the same set of RBridges or links as a packet sent from RB2 to RB1. Even within a given flow, packets from RB1 to RB2 do not necessarily traverse the same path as packets from RB2 to RB1.

3.2.1. One-Way Performance Monitoring

In one-way PM, RB1 sends PM messages to RB2, allowing RB2 to monitor the performance on the path from RB1 to RB2.

A MEP that implements TRILL PM SHOULD support one-way performance monitoring. A MEP that implements TRILL PM SHOULD support both the PM functionality of the sender, RB1, and the PM functionality of the receiver, RB2.

One-way PM can be applied either proactively or on-demand, although the more typical scenario is the proactive mode, where RB1 and RB2 periodically transmit PM messages to each other, allowing each of them to monitor the performance on the incoming path from the peer MEP.

3.2.2. Two-Way Performance Monitoring

In two-way PM, a sender, RB1, sends PM messages to a reflector, RB2, and RB2 responds to these messages, allowing RB1 to monitor the performance of:

- o The path from RB1 to RB2.
- o The path from RB2 to RB1.
- o The two-way path from RB1 to RB2, and back to RB1.

Note that in some cases it may be interesting for RB1 to monitor only the path from RB1 to RB2. Two-way PM allows the sender, RB1, to monitor the path from RB1 to RB2, as opposed to one-way PM (Section 3.2.1.), which allows the receiver, RB2, to monitor this path.

A MEP that implements TRILL PM MUST support two-way PM. A MEP that implements TRILL PM MUST support both the sender and the reflector PM functionality.

As described in Section 3.1. , flow-based PM uses the Flow Entropy field as one of the parameters that identify a flow. In two-way PM,

the Flow Entropy of the path from RB1 to RB2 is typically different from the Flow Entropy of the path from RB2 to RB1. This document uses the Reflector Entropy TLV [TRILL-FM], which allows the sender to specify the Flow Entropy value to be used in the response message.

Two-way PM can be applied either proactively or on-demand.

3.3. Point-to-point vs. Point-to-multipoint PM

PM can be applied either as a point-to-point measurement protocol, or as a point-to-multi-point measurement protocol.

The point-to-point approach measures the performance between two RBridges using unicast PM messages.

In the point-to-multipoint approach, an RBridge RB1 sends PM messages to multiple RBridges using multicast messages. The reflectors (in two-way PM) respond to RB1 using unicast messages. To protect against reply storms, the reflectors MUST send the response messages after a random delay in the range of 0 to 2 seconds. This ensures that the responses are staggered in time, and that the initiating RBridge is not overwhelmed with responses. Moreover, a scope TLV [TRILL-FM] can be used to limit the set of RBridges from which a response is expected, thus reducing the impact of potential response bursts.

4. Loss Measurement

The Loss Measurement protocol has two flavors, one-way Loss Measurement, and two-way Loss Measurement.

Note: The terms 'one-way' and 'two-way' Loss Measurement should not be confused with the terms 'single-ended' and 'dual-ended' Loss Measurement used in [Y.1731]. As defined in Section 3.2., the terms 'one-way' and 'two-way' specify whether the protocol monitors performance on one direction, or on both directions. The terms 'single-ended' and 'dual-ended', on the other hand, describe whether the protocol is asymmetric or symmetric, respectively.

4.1. One-Way Loss Measurement

One-way Loss Measurement measures the one-way packet loss from one MEP to another. The loss ratio is measured using a set of One-way Synthetic Loss Measurement (1SL) messages. The packet format of the 1SL message is specified in Section 6.2.2. Figure 2 illustrates a one-way Loss Measurement message exchange.

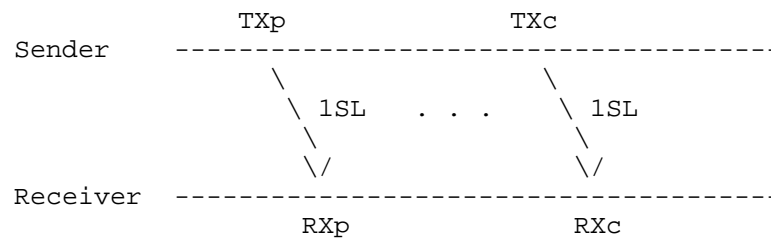


Figure 2 One-Way Loss Measurement

The one-way Loss Measurement procedure uses a set of 1SL messages to measure the packet loss. The figure shows two non-consecutive messages from the set.

The sender maintains a counter of transmitted 1SL messages, and includes the value of this counter, TX, in each 1SL message it transmits. The receiver maintains a counter of received 1SL messages, RX, and can calculate the loss by comparing its counter values to the counter values received in the 1SL messages.

In Figure 2, the subscript 'c' is an abbreviation for current, and 'p' is an abbreviation for previous.

4.1.1.1. 1SL Message Transmission

One-way Loss Measurement can be applied either proactively or on-demand, although as mentioned in Section 3.2.1. , it is more likely to be applied proactively.

The term 'on-demand' in the context of one-way Loss Measurement implies that the sender transmits a fixed set of 1SL messages, allowing the receiver to perform the measurement based on this set.

A MEP that supports one-way Loss Measurement MUST support unicast transmission of 1SL messages.

A MEP that supports one-way Loss Measurement MAY support multicast transmission of 1SL messages.

The sender MUST maintain a packet counter for each peer MEP and probe instance (test ID). Every time the sender transmits a 1SL packet, it

increments the corresponding counter, and then integrates the value of the counter into the <Counter TX> field of the 1SL packet.

The 1SL message MAY be sent with a variable size Data TLV, allowing loss measurement for various packet sizes.

4.1.2. 1SL Message Reception

The receiver MUST maintain a reception counter for each peer MEP and probe instance (test ID). Upon receiving a 1SL packet, the receiver MUST verify that:

- o The 1SL packet is destined to the current MEP.
- o The packet's MD level matches the MEP's MD level.

If both conditions are satisfied, the receiver increments the corresponding receive packet counter, and records the new value of the counter, RX1.

A MEP that supports one-way Loss Measurement MUST support reception of both unicast and multicast 1SL messages.

The receiver computes the one-way packet loss with respect to a probe instance measurement interval. A probe instance measurement interval includes a sequence of 1SL messages with the same test ID. The one-way packet loss is computed by comparing the counter values TXp and RXp at the beginning of the measurement interval, and the counter values TXc and RXc at the end of the measurement interval (Figure 2):

$$\text{one-way packet loss} = (\text{TXc} - \text{TXp}) - (\text{RXc} - \text{RXp}) \quad (1)$$

The calculation in Equation (1) is based on counter value differences, implying that the sender's counter, TX, and the receiver's counter, RX, are not required to be synchronized with respect to a common initial value.

It is noted that if the sender or receiver resets one of the counters, TX or RX, the calculation in Equation (1) produces a false measurement result. Hence the sender and receiver SHOULD NOT clear the TX and RX counters during a measurement interval.

When the receiver calculates the packet loss per Equation (1) it MUST perform a wraparound check. If the receiver detects that one of the counters has wrapped around, the receiver adjusts the result of Equation (1) accordingly.

A 1SL receiver MUST support reception of 1SL messages with a Data TLV.

Since synthetic one-way Loss Measurement is performed using 1SL messages, obviously some 1SL messages may be dropped during a measurement interval. Thus, when the receiver does not receive a 1SL, the receiver cannot perform the calculations in Equation (1) for that specific 1SL message.

4.2. Two-Way Loss Measurement

Two-way Loss Measurement allows a MEP to measure the packet loss on the paths to and from a peer MEP. Two-way Loss Measurement uses a set of Synthetic loss Measurement Messages (SLM) to compute the packet loss. Each SLM is answered with a Synthetic loss Measurement Reply (SLR). The packet formats of the SLM and SLR packets are specified in Sections 6.2.3. and 6.2.4. , respectively. Figure 2 illustrates a two-way Loss Measurement message exchange.

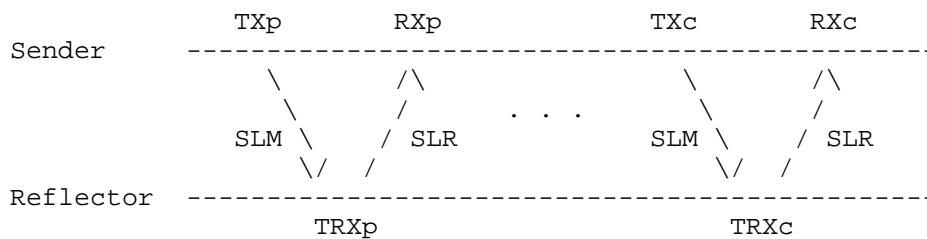


Figure 3 Two-Way Loss Measurement

The two-way Loss Measurement procedure uses a set of SLM-SLR handshakes. The figure shows two non-consecutive handshakes from the set.

The sender maintains a counter of transmitted SLM messages, and includes the value of this counter, TX, in each transmitted SLM message. The reflector maintains a counter of received SLM messages, TRX. The reflector generates an SLR, and incorporates TRX into the SLR packet. The sender maintains a counter of received SLR messages,

RX. Upon receiving an SLR message, the sender can calculate the loss by comparing the local counter values to the counter values received in the SLR messages.

The subscript 'c' is an abbreviation for current, and 'p' is an abbreviation for previous.

4.2.1. SLM Message Transmission

Two-way Loss Measurement can be applied either proactively or on-demand.

A MEP that supports two-way Loss Measurement MUST support unicast transmission of SLM messages.

A MEP that supports two-way Loss Measurement MAY support multicast transmission of SLM messages.

The sender MUST maintain a counter of transmitted SLM packets for each peer MEP and probe instance (test ID). Every time the sender transmits an SLM packet it increments the corresponding counter, and then integrates the value of the counter into the <Counter TX> field of the SLM packet.

A sender MAY include a Reflector Entropy TLV in an SLM message. The Reflector Entropy TLV format is specified in [TRILL-FM].

An SLM message MAY be sent with a Data TLV, allowing loss measurement for various packet sizes.

4.2.2. SLM Message Reception

The reflector MUST maintain a reception counter, TRX, for each peer MEP and probe instance (test ID).

Upon receiving an SLM packet, the reflector MUST verify that:

- o The SLM packet is destined to the current MEP.
- o The packet's MD level matches the MEP's MD level.

If both conditions are satisfied, the reflector increments the corresponding packet counter, and records the value of the new counter, TRX. The reflector then generates an SLR message that is identical to the received SLM, except for the following modifications:

- o The reflector incorporates TRX into the <Counter TRX> field of the SLR.
- o The <OpCode> field in the OAM header is set to the SLR OpCode.
- o The reflector assigns its MEP ID in the <Reflector MEP ID> field.
- o If the received SLM includes a Reflector Entropy TLV [TRILL-FM], the reflector copies the value of the Flow Entropy from the TLV into the <Flow Entropy> field of the SLR message. The outgoing SLR message does not include a Reflector Entropy TLV.
- o The TRILL header and transport header are modified to reflect the source and destination of the SLR packet. The SLR is always a unicast message.

A MEP that supports two-way Loss Measurement MUST support reception of both unicast and multicast SLM messages.

A reflector MUST support reception of SLM packets with a Data TLV. When receiving an SLM with a Data TLV, the reflector includes the unmodified TLV in the SLR.

4.2.3. SLR Message Reception

The sender MUST maintain a reception counter, RX, for each peer MEP and probe instance (test ID).

Upon receiving an SLR message, the sender MUST verify that:

- o The SLR packet is destined to the current MEP.
- o The <Sender MEP ID> field in the SLR packet matches the current MEP.
- o The packet's MD level matches the MEP's MD level.

If the conditions above are met, the sender increments the corresponding reception counter, and records the new value, RX.

The sender computes the packet loss with respect to a probe instance measurement interval. A probe instance measurement interval includes a sequence of SLM messages, and their corresponding SLR messages, all with the same test ID. The packet loss is computed by comparing the counters at the beginning of the measurement interval, denoted with a subscript 'p', and the counters at the end of the measurement interval, denoted with a subscript 'c' (as illustrated in Figure 3).

$$\text{far-end packet loss} = (\text{TXc-TXp}) - (\text{TRXc-TRXp}) \quad (2)$$

$$\text{near-end packet loss} = (\text{TRXc-TRXp}) - (\text{RXc-RXp}) \quad (3)$$

Note: total two-way packet loss is the sum of the far and near end packet losses, that is $(\text{TXc-TXp}) - (\text{RXc-RXp})$.

The calculations in the two equations above are based on counter value differences, implying that the sender's counters, TX and RX, and the reflector's counter, TRX, are not required to be synchronized with respect to a common initial value.

It is noted that if the sender or reflector resets one of the counters, TX, TRX or RX, the calculation in Equations (2) and (3) produces a false measurement result. Hence the sender and reflector SHOULD NOT clear the TX, TRX and RX counters during a measurement interval.

When the sender calculates the packet loss per Equations (2) and (3) it MUST perform a wraparound check. If the reflector detects that one of the counters has wrapped around, the reflector adjusts the result of Equations (2) and (3) accordingly.

Since synthetic two-way Loss Measurement is performed using SLM and SLR messages, obviously some SLM and SLR messages may be dropped during a measurement interval. When an SLM or an SLR is dropped, the corresponding two-way handshake (Figure 3) is not completed successfully, and thus the reflector does not perform the calculations in Equations (2) and (3) for that specific message exchange.

A sender MAY choose to monitor only the far-end packet loss, that is, perform the computation in Equation (2), and ignore the computation in Equation (3). Note that, in this case, the sender can run flow-based PM of the path TO the peer MEP without using the Reflector Entropy TLV.

5. Delay Measurement

The Delay Measurement protocol has two flavors, One-Way Delay Measurement, and Two-Way Delay Measurement.

5.1. One-Way Delay Measurement

One-way Delay Measurement is used for computing the one-way packet delay from one MEP to another. The packet format used in one-way Delay Measurement is referred to as LDM, and is specified in Section

6.3.2. The one-way Delay Measurement message exchange is illustrated in Figure 4.

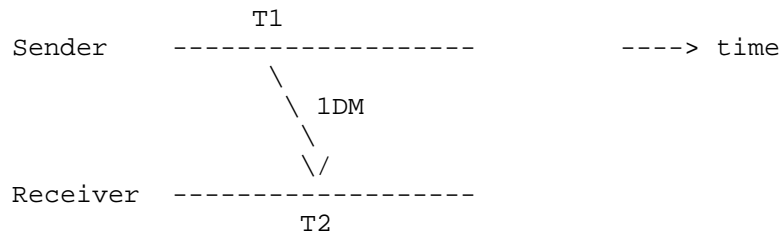


Figure 4 One-Way Delay Measurement

The sender transmits a 1DM message incorporating its time of transmission, T1. The receiver then receives the message at time T2, and calculates the one-way delay as:

$$\text{one-way delay} = T2 - T1 \quad (4)$$

Equation (4) implies that T2 and T1 are measured with respect to a common reference time. Hence, two MEPs running an one-way Delay Measurement protocol MUST be time-synchronized. The method used for synchronizing the clocks associated with the two MEPs is outside the scope of this document.

5.1.1. 1DM Message Transmission

1DM packets can be transmitted proactively or on-demand, although as mentioned in Section 3.2.1. , they are typically transmitted proactively.

A MEP that supports one-way Delay Measurement MUST support unicast transmission of 1DM messages.

A MEP that supports one-way Delay Measurement MAY support multicast transmission of 1DM messages.

A 1DM message MAY be sent with a variable size Data TLV, allowing packet delay measurement for various packet sizes.

The sender incorporates the 1DM packet's time of transmission into the <Timestamp T1> field.

5.1.2. 1DM Message Reception

Upon receiving a 1DM packet, the receiver records its time of reception, T_2 . The receiver **MUST** verify two conditions:

- o The 1DM packet is destined to the current MEP.
- o The packet's MD level matches the MEP's MD level.

If both conditions are satisfied, the receiver terminates the packet and calculates the one-way delay as specified in Equation (4).

A MEP that supports one-way Delay Measurement **MUST** support reception of both unicast and multicast 1DM messages.

A 1DM receiver **MUST** support reception of 1DM messages with a Data TLV.

When one-way Delay Measurement packets are received periodically, the receiver **MAY** compute the packet delay variation based on multiple measurements. Note that packet delay variation can be computed even when the two peer MEPs are not time synchronized.

5.2. Two-Way Delay Measurement

Two-way Delay Measurement uses a two-way handshake for computing the two-way packet delay between two MEPs. The handshake includes two packets, a Delay Measurement Message (DMM) and a Delay Measurement Reply (DMR). The DMM and DMR packet formats are specified in Section 6.3.3. and 6.3.4. , respectively.

The two-way Delay Measurement message exchange is illustrated in Figure 5.

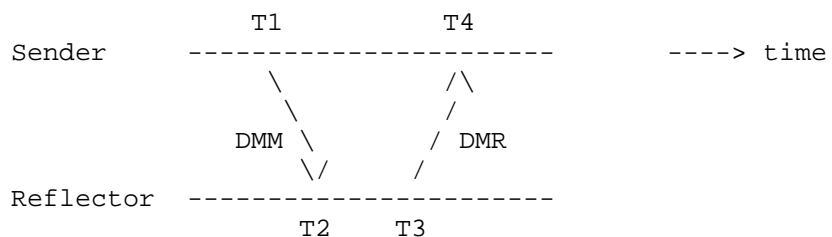


Figure 5 Two-Way Delay Measurement

The sender generates a DMM message incorporating its time of transmission, T1. The reflector receives the DMM message and records its time of reception, T2. The reflector then generates a DMR message, incorporating T1, T2 and the DMR's transmission time, T3. The sender receives the DMR message at T4, and using the 4 timestamps it calculates the two-way packet delay.

5.2.1. DMM Message Transmission

DMM packets can be transmitted periodically or on-demand.

A MEP that supports two-way Delay Measurement **MUST** support unicast transmission of DMM messages.

A MEP that supports two-way Delay Measurement **MAY** support multicast transmission of DMM messages.

A sender **MAY** include a Reflector Entropy TLV in a DMM message. The Reflector Entropy TLV format is specified in [TRILL-FM].

A DMM **MAY** be sent with a variable size Data TLV, allowing packet delay measurement for various packet sizes.

The sender incorporates the DMM packet's time of transmission into the <Timestamp T1> field.

5.2.2. DMM Message Reception

Upon receiving a DMM packet, the reflector records its time of reception, T2. The reflector **MUST** verify two conditions:

- o The DMM packet is destined to the current MEP.
- o The packet's MD level matches the MEP's MD level.

If both conditions are satisfied, the reflector terminates the packet, and generates a DMR packet. The DMR is identical to the received DMM, except for the following modifications:

- o The reflector incorporates T2 into the <Timestamp T2> field of the DMR.
- o The reflector incorporates the DMR's transmission time, T3, into the <Timestamp T3> field of the DMR.

- o The <OpCode> field in the OAM header is set to the DMR OpCode.
- o If the received DMM includes a Reflector Entropy TLV [TRILL-FM], the reflector copies the value of the Flow Entropy from the TLV into the <Flow Entropy> field of the DMR message. The outgoing DMR message does not include a Reflector Entropy TLV.
- o The TRILL header and transport header are modified to reflect the source and destination of the DMR packet. The DMR is always a unicast message.

A MEP that supports two-way Delay Measurement MUST support reception of both unicast and multicast DMM messages.

A reflector MUST support reception of DMM packets with a Data TLV. When receiving a DMM with a Data TLV, the reflector includes the unmodified TLV in the DMR.

5.2.3. DMR Message Reception

Upon receiving the DMR message, the sender records its time of reception, T4. The sender MUST verify:

- o The DMR packet is destined to the current MEP.
- o The packet's MD level matches the MEP's MD level.

If both conditions above are met, the sender uses the 4 timestamps to compute the two-way delay:

$$\text{two-way delay} = (T4 - T1) - (T3 - T2) \quad (5)$$

Note that two-way delay can be computed even when the two peer MEPs are not time synchronized. One-way Delay Measurement, on the other hand, requires the two MEPs to be synchronized.

Two MEPs running a two-way Delay Measurement protocol MAY be time-synchronized. If two-way Delay Measurement is run between two time-synchronized MEPs, the sender MAY compute the one-way delays:

$$\text{one-way delay \{sender->reflector\}} = T2 - T1 \quad (6)$$

$$\text{one-way delay \{reflector->sender\}} = T4 - T3 \quad (7)$$

When two-way Delay Measurement is run periodically, the sender MAY also compute the delay variation based on multiple measurements.

A sender MAY choose to monitor only the sender->reflector delay, that is, perform the computation in Equation (6), and ignore the computations in (5) and (7). Note that in this case the sender can run flow-based PM of the path to the peer MEP without using the Reflector Entropy TLV.

6. Packet Formats

6.1. TRILL OAM Encapsulation

The TRILL OAM packet format is generally discussed in [OAM-FRAMEWK], and specified in detail in [TRILL-FM]. It is quoted in this document for convenience.



[Page 20]

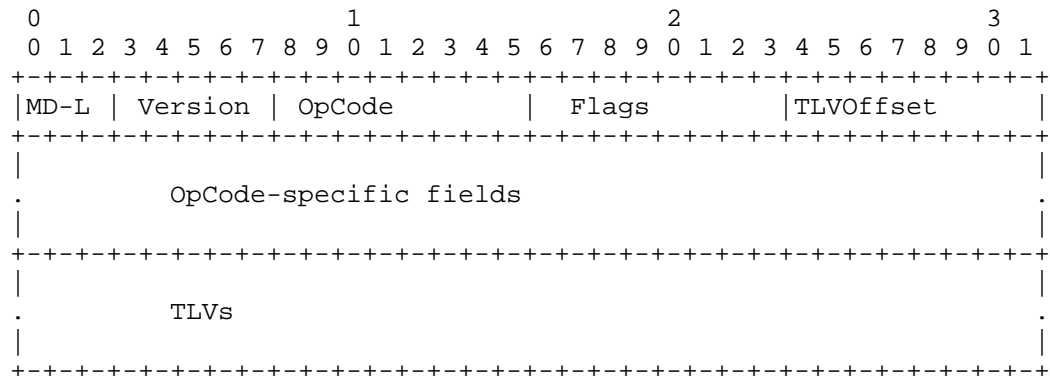


Figure 7 OAM Packet Format

The first 4 octets of the OAM Message Channel are common to all OpCodes, whereas the rest is OpCode-specific. Below is a brief summary of the fields in the first 4 octets:

- o MD-L : Maintenance Domain Level.
- o Version: indicates the version of this protocol. Always zero in the context of this document.
- o Flags: always zero in the context of this document.
- o FirstTLVOffset: defines the location of the first TLV, in octets, starting from the end of the FirstTLVOffset field.

For further details about the OAM packet format, see [TRILL-FM].

6.2. Loss Measurement Packet Formats

6.2.1. Counter Format

Loss Measurement packets use a 32-bit packet counter field. When a counter is incremented beyond its maximal value, 0xFFFFFFFF, it wraps around back to 0.

6.2.2. 1SL Packet Format

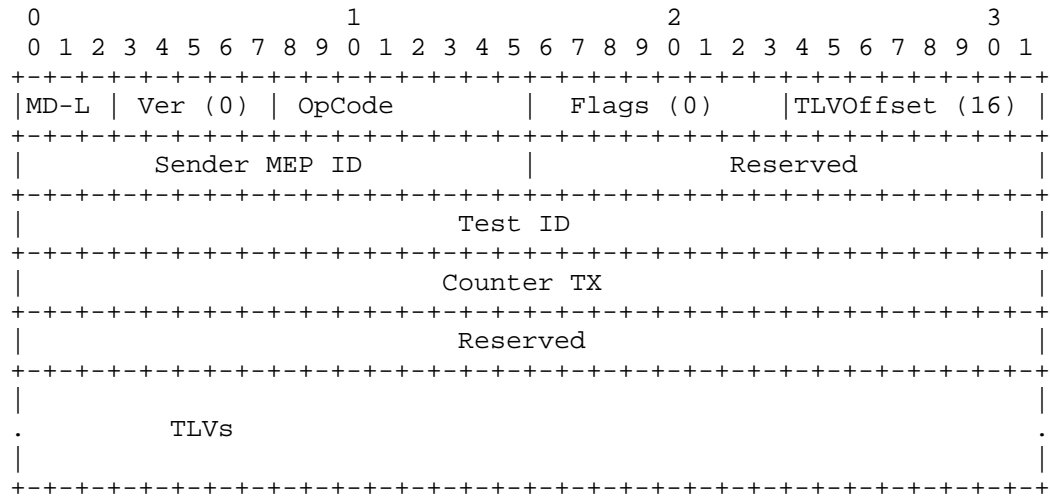


Figure 8 1SL Packet Format

- o Sender MEP ID: the MEP ID of the MEP that initiated the 1SL.
- o Reserved: always 0.
- o Test ID: a 32-bit unique test identifier.
- o Counter TX: the value of the sender's transmission counter, including this packet, at the time of transmission.

6.2.3. SLM Packet Format

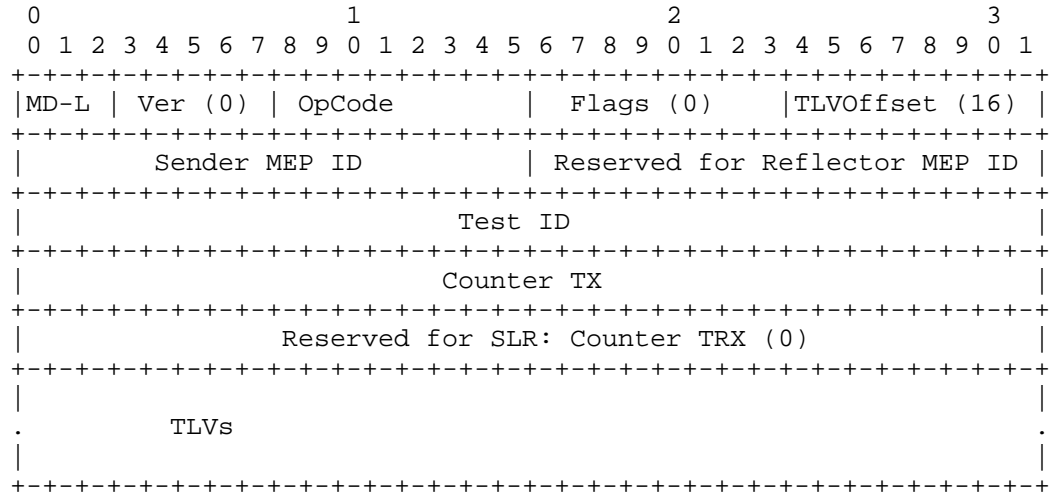


Figure 9 SLM Packet Format

- o Sender MEP ID: the MEP ID of the MEP that initiated this packet.
- o Reserved: this field is reserved for the reflector's MEP ID, to be added in the SLR.
- o Test ID: a 32-bit unique test identifier.
- o Counter TX: the value of the sender's transmission counter, including this packet, at the time of transmission.
- o Reserved: this field is reserved for the SLR corresponding to this packet. The reflector uses this field in the SLR for carrying TRX, the value of its reception counter.

6.2.4. SLR Packet Format

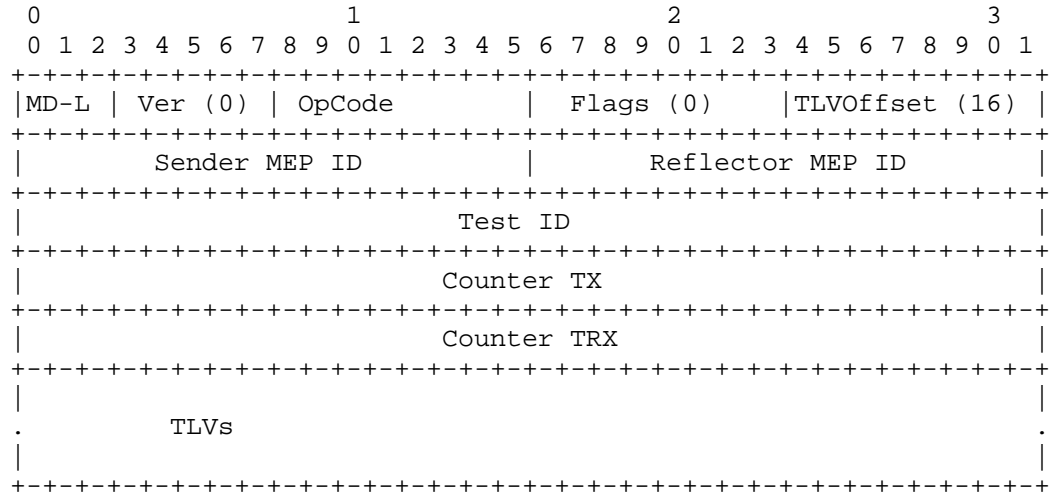


Figure 10 SLR Packet Format

- o Sender MEP ID: the MEP ID of the MEP that initiated the SLM that this SLR replies to.
- o Reflector MEP ID: the MEP ID of the MEP that transmits this SLR message.
- o Test ID: a 32-bit unique test identifier, copied from the corresponding SLM message.
- o Counter TX: the value of the sender's transmission counter at the time of the SLM transmission.
- o Counter TRX: the value of the reflector's reception counter, including this packet, at the time of reception of the corresponding SLM packet.

6.3. Delay Measurement Packet Formats

6.3.1. Timestamp Format

The timestamps used in Delay Measurement packets are 64 bits long. These timestamps use the 64 least significant bits of the IEEE 1588-2008 (1588v2) Precision Time Protocol timestamp format [IEEE1588].

This truncated format consists of a 32-bit seconds field followed by a 32-bit nanoseconds field. This truncated format is also used in IEEE 1588v1, in [Y.1731], and in [MPLS-LM-DM].

6.3.2. 1DM Packet Format

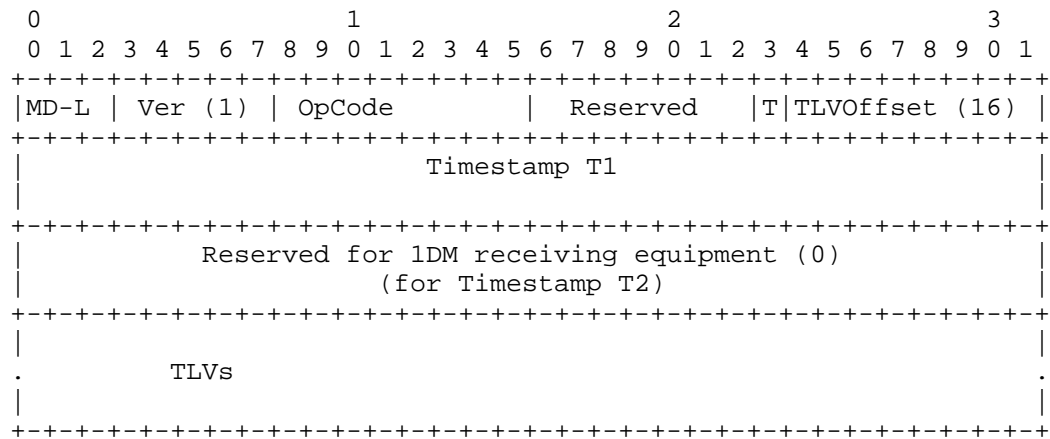


Figure 11 1DM Packet Format

- o T: Type flag. When this flag is set it indicates proactive operation, and when cleared it indicates on-demand mode.
- o Timestamp T1: specifies the time of transmission of this packet.
- o Reserved: this field is reserved for internal usage of the 1DM receiver. The receiver can use this field for carrying T2, the time of reception of this packet.

6.3.3. DMM Packet Format

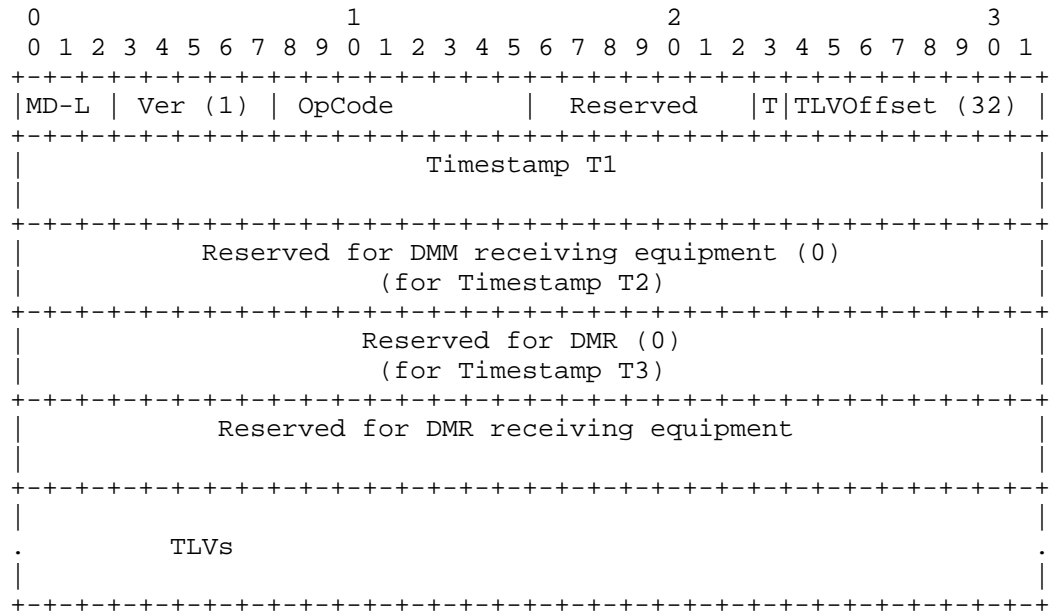


Figure 12 DMM Packet Format

- o T: Type flag. When this flag is set it indicates proactive operation, and when cleared it indicates on-demand mode.
- o Timestamp T1: specifies the time of transmission of this packet.
- o Reserved: this field is reserved for internal usage of the MEP that receives the DMM (the reflector). The reflector can use this field for carrying T2, the time of reception of this packet.
- o Reserved for DMR: two timestamp fields are reserved for the DMR message. One timestamp field is reserved for T3, the DMR transmission time, and the other field is reserved for internal usage of the MEP that receives the DMR.

6.3.4. DMR Packet Format

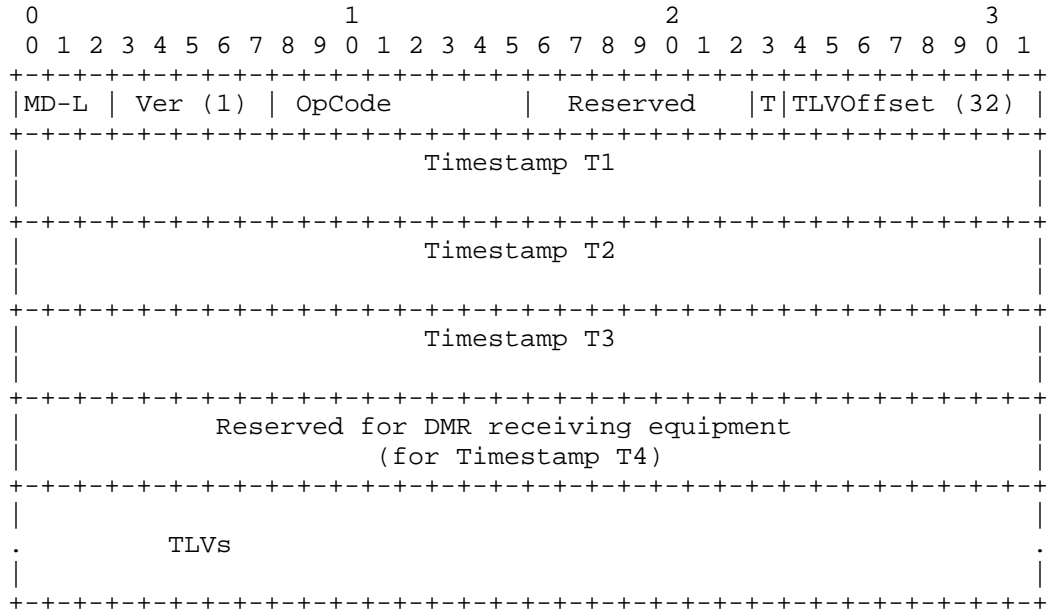


Figure 13 DMR Packet Format

- o **T**: Type flag. When this flag is set it indicates proactive operation, and when cleared it indicates on-demand mode.
- o **Timestamp T1**: specifies the time of transmission of the DMM packet that this DMR replies to.
- o **Timestamp T2**: specifies the time of reception of the DMM packet that this DMR replies to.
- o **Timestamp T3**: specifies the time of transmission of this DMR packet.
- o **Reserved**: this field is reserved for internal usage of the MEP that receives the DMR (the sender). The sender can use this field for carrying T4, the time of reception of this packet.

6.4. OpCode Values

As the OAM packets specified herein generally conform to [Y.1731], the same OpCodes are used as follows:

OpCode value -----	OAM packet type -----
45	1DM
46	DMR
47	DMM
53	1SL
54	SLR
55	SLM

7. Performance Monitoring Process

The Performance Monitoring process is made up of a number of Performance Monitoring instances, known as PM Sessions. A PM session can be initiated between two MEPs on a specific flow and be defined as either a Loss Measurement session or Delay Measurement session.

The Loss Measurement session can be used to determine the performance metrics Frame Loss Ratio, availability, and resiliency. The Delay Measurement session can be used to determine the performance metrics Frame Delay, Inter-Frame Delay Variation, Frame Delay Range, and Mean Frame Delay.

The PM session is defined by the specific PM function (PM tool) being run, and also by the Start Time, Stop time, Message Period, Measurement Interval, and Repetition Time. These terms are defined as follows:

- o The Start Time is the time that the PM session begins.
- o The Stop Time is the time that the measurement ends.
- o The Message Period is the message transmission frequency (the time between message transmissions).

11. References

11.1. Normative References

- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFCTRILL] Perlman, R., Eastlake, D., Dutt, D., Gai, S., Ghanwani, A., "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.
- [OAM-FRAMEWK] Salam, S., Senevirathne, T., Aldrin, S., Eastlake, D., "TRILL OAM Framework", draft-ietf-trill-oam-framework (work in progress), September 2013.
- [RFC-FGL] D. Eastlake, M. Zhang, P. Agarwal, R. Perlman, D. Dutt, "TRILL: Fine-Grained Labeling", draft-ietf-trill-fine-labeling (in RFC Editor's queue), 2014.
- [TRILL-FM] Senevirathne, T., Finn, N., Salam, S., Kumar, D., Eastlake, D., Aldrin, S., Li, Y., "TRILL Fault Management", draft-ietf-trill-oam-fm (work in progress), July 2013.

11.2. Informative References

- [OAM-REQ] Senevirathne, T., Bond, D., Aldrin, S., Li, Y., Watve, R., "Requirements for Operations, Administration and Maintenance (OAM) in Transparent Interconnection of Lots of Links (TRILL)", RFC 6905, March 2013.
- [Y.1731] ITU-T Recommendation G.8013/Y.1731, "OAM Functions and Mechanisms for Ethernet-based Networks", July 2011.
- [802.1Q] "IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q(tm), 2012 Edition, October 2012.
- [IEEE1588] IEEE TC 9 Instrumentation and Measurement Society, "1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Standard, 2008.
- [MPLS-LM-DM] Frost, D., Bryant, S., "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, September 2011.

- [OAM] Andersson, L., Van Helvoort, H., Bonica, R., Romascanu, D., Mansfield, S., "Guidelines for the use of the OAM acronym in the IETF ", RFC 6291, June 2011.
- [IPPM-1DM] Almes, G., Kalidindi, S. and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
- [IPPM-2DM] Almes, G., Kalidindi, S. and M. Zekauskas, "A round-trip delay metric for IPPM", RFC 2681, September 1999.
- [IPPM-Loss] Almes, G., Kalidindi, S. and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.

Authors' Addresses

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692 Israel

Email: talmi@marvell.com

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

Email: tsenevir@cisco.com

Samer Salam
Cisco
595 Burrard Street, Suite 2123
Vancouver, BC V7X 1J1, Canada

Email: ssalam@cisco.com

Deepak Kumar
Cisco
510 McCarthy Blvd,
Milpitas, CA 95035, USA

Phone : +1 408-853-9760
Email: dekumar@cisco.com

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

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

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

Tissa Senevirathne
Norman Finn
Samer Salam
Deepak Kumar
CISCO

Donald Eastlake
Sam Aldrin
Yizhou Li
Huawei

February 13, 2014

Expires: August 2014

TRILL Fault Management
draft-ietf-trill-oam-fm-02.txt

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), 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/ietf/lid-abstracts.txt>

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

This Internet-Draft will expire on August 13, 2014.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

This document specifies TRILL OAM Fault Management. Methods in this document follow the IEEE 802.1 CFM (Continuity Fault Management) framework and reuse OAM tools where possible. Additional messages and TLVs are defined for TRILL specific applications or where a different set of information is required other than IEEE 802.1 CFM. This document updates RFC 6325.

Table of Contents

1. Introduction	4
2. Conventions used in this document	4
3. General Format of TRILL OAM Packets	5
3.1. Identification of TRILL OAM frames	7
3.2. Use of TRILL OAM Alert Flag	7
3.2.1. Handling of TRILL frames with the "A" Flag	8
3.3. OAM Capability Announcement	8
4. TRILL OAM Layering vs. IEEE Layering	9
4.1. Processing at ISS Layer	11
4.1.1. Receive Processing	11
4.1.2. Transmit Processing	11
4.2. End Station VLAN and Priority Processing	11
4.2.1. Receive Processing	11
4.2.2. Transmit Processing	11
4.3. TRILL Encapsulation and De-capsulation Layer	11
4.3.1. Receive Processing for Unicast packets	11
4.3.2. Transmit Processing for unicast packets	12
4.3.3. Receive Processing for Multicast packets	13
4.3.4. Transmit Processing of Multicast packets	14
4.4. TRILL OAM Layer Processing	15
5. Maintenance Associations (MA) in TRILL	16
6. MEP Addressing	17
6.1. Use of MIP in TRILL	20
7. Continuity Check Message (CCM)	22
8. TRILL OAM Message Channel	24

8.1. TRILL OAM Message header	24
8.2. TRILL Specific OAM Opcodes	25
8.3. Format of TRILL OAM TLV	25
8.4. TRILL OAM TLVs	26
8.4.1. Common TLVs between CFM and TRILL	26
8.4.2. TRILL OAM Specific TLVs	27
8.4.3. TRILL OAM Application Identifier TLV	27
8.4.4. Out Of Band Reply Address TLV	28
8.4.5. Diagnostics Label TLV	29
8.4.6. Original Data Payload TLV	30
8.4.7. RBridge scope TLV	31
8.4.8. Previous RBridge nickname TLV	32
8.4.9. Next Hop RBridge List TLV	32
8.4.10. Multicast Receiver Port count TLV	33
8.4.11. Flow Identifier (flow-id) TLV	34
8.4.12. Reflector Entropy TLV	34
8.4.13. OAM Authentication TLV	35
9. Loopback Message	36
9.1. Loopback OAM Message format	36
9.2. Theory of Operation	37
9.2.1. Actions by Originator RBridge	37
9.2.2. Intermediate RBridge	37
9.2.3. Destination RBridge	38
10. Path Trace Message	38
10.1. Theory of Operation	39
10.1.1. Action by Originator RBridge	39
10.1.2. Intermediate RBridge	40
10.1.3. Destination RBridge	41
11. Multi-Destination Tree Verification (MTV) Message	41
11.1. Multi-Destination Tree Verification (MTV) OAM Message Format	42
11.2. Theory of Operation	42
11.2.1. Actions by Originator RBridge	42
11.2.2. Receiving RBridge	43
11.2.3. In scope RBridges	43
12. Application of Continuity Check Message (CCM) in TRILL ...	44
12.1. CCM Error Notification	45
12.2. Theory of Operation	46
12.2.1. Actions by Originator RBridge	46
12.2.2. Intermediate RBridge	47
12.2.3. Destination RBridge	47
13. Fragmented Reply	48
14. Security Considerations	48
15. IANA Considerations	49
15.1. OAM Capabilitiy Flags	49
15.2. CFM Code Points	50
15.3. MAC Addresses	50

15.4. Return codes and sub codes	51
16. References	51
16.1. Normative References	51
16.2. Informative References	51
17. Acknowledgments	52
Appendix A. Backwards Compatibility	54
Appendix B. Base Mode for TRILL OAM	57
Appendix C. Unicast MAC Request	59

1. Introduction

The general structure of TRILL OAM messages is presented in [TRLOAMFRM]. TRILL OAM messages consist of five parts: link header, TRILL header, flow entropy, OAM message channel, and link trailer.

The OAM message channel carries various control information and OAM related data between TRILL switches, also known as RBridges or Routing Bridges.

A common OAM message channel representation can be shared between different technologies. This consistency between different OAM technologies promotes nested fault monitoring and isolation between technologies that share the same OAM framework.

The TRILL OAM message channel is formatted as specified in IEEE Connectivity Fault Management (CFM) [8021Q].

The ITU-T Y.1731 [Y1731] standard utilizes the same messaging format as [8021Q] OAM messages where applicable. This document takes a similar stance and reuses [8021Q] in TRILL OAM. It is assumed readers are familiar with [8021Q] and [Y1731]. Readers who are not familiar with these documents are encouraged to review them.

This document updates [RFC6325] as specified in Section 3.1.

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].

Acronyms used in the document include the following:

CCM - Continuity Check Message [8021Q]
ECMP - Equal Cost Multipath
ISS - Internal Sub Layer Service [8021Q]
LBM - Loop Back Message [8021Q]
MP - Maintenance Point [TRLOAMFRM]
MEP - Maintenance End Point [TRLOAMFRM] [8021Q]
MIP - Maintenance Intermediate Point [TRLOAMFRM] [8021Q]
MA - Maintenance Association [8021Q] [TRLOAMFRM]
MD - Maintenance Domain [8021Q]
MTV - Multi-destination Tree Verification Message
OAM - Operations, Administration, and Maintenance [RFC6291]
PRI - Priority of Ethernet Frames [8021Q]
PTM - Path Trace Message
TRILL - Transparent Interconnection of Lots of Links [RFC6325]
SAP - Service Access Point [8021Q]

3. General Format of TRILL OAM Packets

The TRILL forwarding paradigm allows an implementation to select a path from a set of equal cost paths to forward a unicast TRILL Data packet. For multi-destination TRILL Data packets, a distribution tree is chosen by the TRILL switch that ingresses or creates the packet. Selection of the path of choice is implementation dependent at each hop for unicast and at the ingress for multi-destination. However, it is a common practice to utilize Layer 2 through Layer 4 information in the frame payload for path selection.

For accurate monitoring and/or diagnostics, OAM Messages are required to follow the same path as corresponding data packets. [TRLOAMFRM] presents the high-level format of the OAM messages. The details of the TRILL OAM frame format are defined in this document.

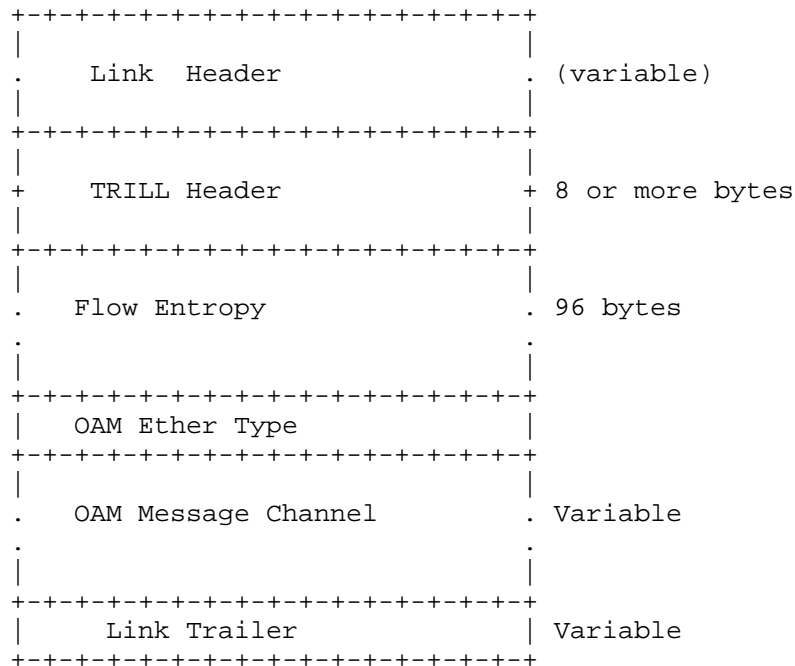


Figure 1 Format of TRILL OAM Messages

Link Header: Media-dependent header. For Ethernet, this includes Destination MAC, Source MAC, VLAN (optional) and EtherType fields.

TRILL Header: Fixed size of 8 bytes when the Extended Header is not included [RFC6325]

Flow Entropy: This is a 96-byte fixed size field. The rightmost bits of the field MUST be padded with zeros, up to 96 bytes, when the flow entropy is less than 96 bytes. Flow entropy enables emulation of the forwarding behavior of the desired data packets.

The Flow Entropy field starts with the Inner.MacDA. The offset of the Inner.MacDA depends on whether extensions are included or not as specified in [TRILLEXT] and [RFC6325]. Such extensions are not commonly supported in current TRILL implementations.

OAM Ether Type: OAM Ether Type is 16-bit EtherType that identifies the OAM Message channel that follows. This document specifies using the EtherType 0x8902 allocated for CFM [8021Q].
OAM Message Channel: This is a variable size section that carries OAM related information. The message format is as specified in [8021Q].

Link Trailer: Media-dependent trailer. For Ethernet, this is the FCS (Frame Check Sequence).

3.1. Identification of TRILL OAM frames

TRILL, as originally specified in [RFC6325], did not have a specific flag or a method to identify OAM frames. This document updates [RFC6325] to include specific methods to identify TRILL OAM frames. Section 3.2. below explains the details of the method.

3.2. Use of TRILL OAM Alert Flag

The TRILL Header, as defined in [RFC6325], has two reserved bits. This document specifies use of the reserved bit next to Version field in the TRILL header as the Alert flag. Alert flag will be denoted by "A". R Bridges MUST NOT use the "A" flag for forwarding decisions such as the selection of which ECMP path or multi-destination tree to select.

Implementations that comply with this document MUST utilize "A" flag and CFM EtherType to identify TRILL OAM frames.

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| V |A|R|M|Op-Length| Hop Count |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Egress RBridge Nickname   |   Ingress RBridge Nickname   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Options...               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 2 TRILL Header with the "A" Flag

A (1 bit) - Indicates this is a possible OAM frame and is subject to specific handling as specified in this document.

All other TRILL Header fields carry the same meaning as defined in RFC6325.

3.2.1. Handling of TRILL frames with the "A" Flag

Value "1" in the A flag indicates TRILL frames that may qualify as OAM frames. Implementations are further REQUIRED to validate such frames by comparing the value at the OAM Ether Type (Figure 1) location with the CFM EtherType "0x8902" [8021Q]. If the value matches, such frames are identified as TRILL OAM frames and SHOULD be processed as discussed in Section 4.

Frames with the "A" flag set that do not contain CFM EtherType are not considered as OAM frames. Such frames MUST be discarded.

3.3. OAM Capability Announcement

Any given RBridge can be (1) OAM incapable or (2) OAM capable with new extensions or (3) OAM capable with backwards-compatible method. The OAM request originator, prior to origination of the request is required to identify the OAM capability of the target and generate the appropriate OAM message.

Capability flags defined in TRILL version sub-TLV (TRILL-VER) [rfc6326bis] will be utilized for announcing OAM capabilities. The following OAM related capability flags are defined:

O - OAM Capable

B - Backwards Compatible OAM

A capability announcement, with "O" Flag set to 1 and "B" flag set to 1, indicates that the originating RBridge is OAM capable but utilizes the backwards compatible method defined in Appendix A. A capability announcement with "O" Flag set to 1 and "B" flag set to 0, indicates that the originating RBridge is OAM capable and utilizes the method specified in section 3.2.

When "O" Flag is set to 0, the announcing implementation is considered not capable of OAM and the "B" flag is ignored.

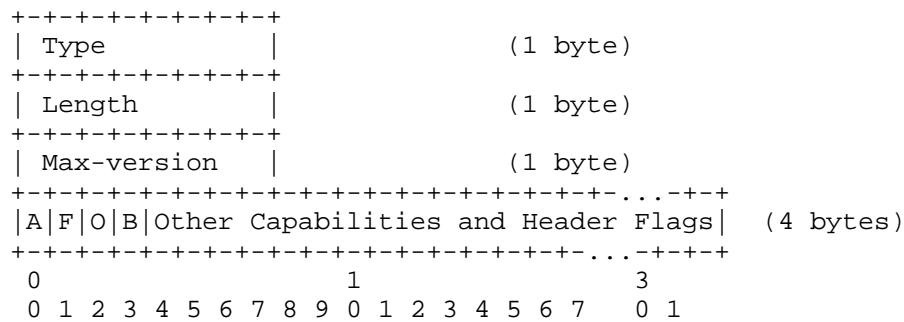


Figure 3 TRILL-VER sub-TLV [rfc6326bis] with O and B flags

Capability flags "A" and "B" are defined by [rfc6326bis] and [rfcFGL]. "O" and "F" Flags are located after "F" flag in the Capability and Header Flags field of TRILL-VER sub-TLV, as depicted in Figure 3 above. Usage of "O" and "B" bits flags discussed above.

4. TRILL OAM Layering vs. IEEE Layering

This section presents the placement of the TRILL OAM shim within the IEEE 802.1 layers. The Transmit and Receive processing are explained.

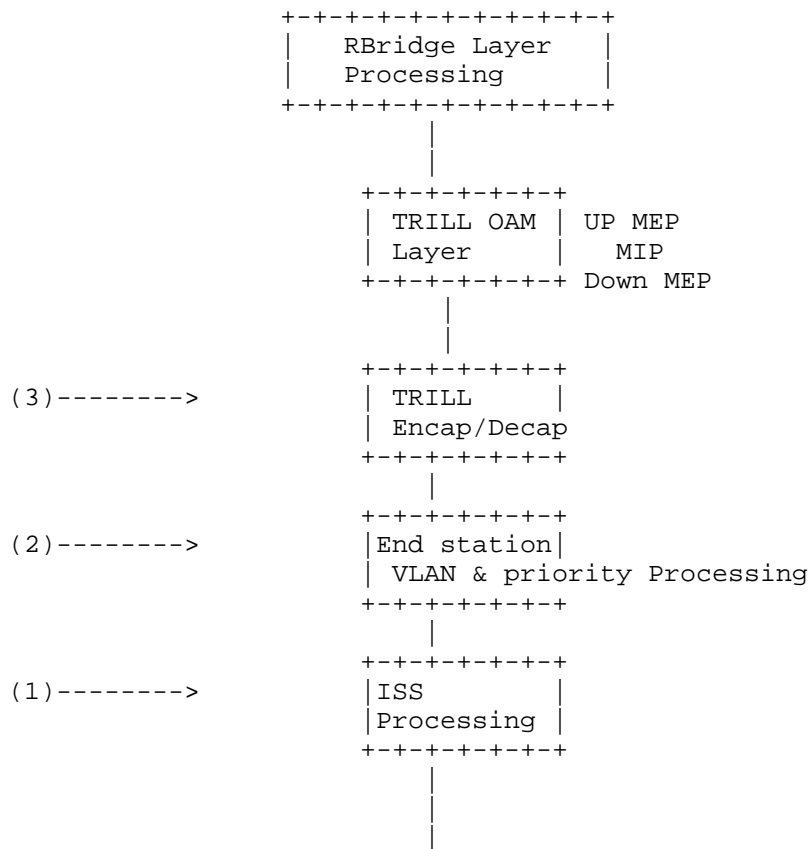


Figure 4 Placement of TRILL MP within IEEE 802.1

[RFC6325] Section 4.6 as updated by [RFCc1correct] provides a detailed explanation of frame processing. Please refer to those documents for additional details and for processing scenarios not covered herein.

Sections 4.1 and 4.2 below apply to links using a broadcast LAN technology such as Ethernet.

On links using an inherently point-to-point technology, such as PPP [RFC6361], there is no Outer.MacDA, Outer.MacSA, or Outer.VLAN because these are part of the link header for

Ethernet. Point-to-point links typically have link headers without these fields.

4.1. Processing at ISS Layer

4.1.1. Receive Processing

The ISS Layer receives an indication from the port. It extracts DA, SA and marks the remainder of the payload as M1. ISS Layer passes on (DA, SA, M1) as an indication to the higher layer.

For TRILL Ethernet frames, this is Outer.MacDA and Outer.MacSA. M1 is the remainder of the packet.

4.1.2. Transmit Processing

The ISS layer receives an indication from the higher layer that contains (DA, SA, M1). It constructs an Ethernet frame and passes down to the port.

4.2. End Station VLAN and Priority Processing

4.2.1. Receive Processing

Receives (DA, SA, M1) indication from ISS Layer. Extracts the VLAN ID and priority from the M1 part of the received indication (or derive them from the port defaults or the like) and constructs (DA, SA, VLAN, PRI, M2). VLAN+PRI+M2 map to M1 in the received indication. Pass (DA, SA, VLAN, PRI, M2) to the TRILL encap/decap procession layer.

4.2.2. Transmit Procession

Receive (DA, SA, VLAN, PRI, M2) indication from TRILL encap/decap processing layer. Merge VLAN, PRI, M2 to form M1. Pass down (DA, SA, M1) to the ISS processing Layer.

4.3. TRILL Encapsulation and De-capsulation Layer

4.3.1. Receive Processing for Unicast packets

Receive indication (DA, SA, VLAN, PRI, M2) from End Station VLAN and Priority Processing Layer.

- o If DA matches port Local DA and Frame is of TRILL EtherType

- . Discard DA, SA, VLAN, PRI. From M2, derive (TRILL-HDR, iDA, iSA, i-VL, M3)
- . If TRILL nickname is Local and TRILL-OAM Flag is set
 - Pass on to OAM processing
- . Else pass on (TRILL-HDR, iDA, iSA, i-VL, M3) to RBridge Layer
 - o If DA matches port Local DA and EtherType is RBridge-Channel [Channel]
 - . Process as a possible unicast native RBridge Channel packet
 - o If DA matches port Local DA and EtherType is neither TRILL nor RBridge-Channel
 - . Discard packet
 - o If DA does not match and port is Appointed Forwarder for VLAN and EtherType is not TRILL or RBridge-Channel
 - . Insert TRILL-Hdr and send (TRILL-HDR, iDA, iSA,i-VL, M3) indication to RBridge Layer <- This is the TRILL ingress function

4.3.2. Transmit Processing for unicast packets

- o Receive indication (TRILL-HDR, iDA, iSA, iVL, M3) from RBridge Layer
- o If egress TRILL nickname is local
 - o If port is Appointed Forwarder for iVL and the port is not configured as a trunk or p2p port and (TRILL Alert Flag set and OAM EtherType present) then
 - . Strip TRILL-HDR and construct (DA, SA, VLAN, M2) <- This is the TRILL egress function
 - o Else
 - . Discard packet
- o If egress TRILL nickname is not local

- o Insert Outer.MacDA, Outer.MacSA, Outer.VLAN, TRILL EtherType and construct (DA, SA, VLAN, M2). Where M2 is (TRILL-HDR, iDA, iSA, iVL, M)
- o Forward (DA, SA, V, M2) to the VLAN End Station processing Layer.

4.3.3. Receive Processing for Multicast packets

- o Receive (DA, SA, V, M2) from VLAN aware end station processing layer
- o If the DA is All-RBridges and the EtherType is TRILL
 - o Strip DA, SA and V. From M2, extract (TRILL-HDR, iDA, iSA, iVL and M3).
 - o If TRILL Alert Flag is set and OAM EtherType is present at the end of Flow entropy
 - . Perform OAM Processing
 - o Else extract the TRILL header, inner MAC addresses and inner VLAN and pass indication (TRILL-HDR, iDA, iSA, iVL and M3) to TRILL RBridge Layer
- o If the DA is All-IS-IS-RBridges and the EtherType is L2-IS-IS then pass frame up to TRILL IS-IS processing
- o If the DA is All-RBridges or All-IS-IS-RBridges but EtherType is not TRILL or L2-IS-IS respectively
 - o Discard the packet
- o If the EtherType is TRILL but the multicast DA is not All-RBridge or if the EtherType is L2-IS-IS but the multicast Da is not All-IS-IS-RBridges
 - o Discard the packet
- o If DA is All-Edge-RBridges and EtherType is RBridge-Channel [Channel]
 - o Process as a possible multicast native RBridge Channel packet

- o If the DA is in the initial bridging/link protocols block (01-80-C2-00-00-00 to 01-80-C2-00-00-0F) or is in the TRILL block and not assigned for Outer.MacDA use (01-80-C2-00-00-42 to 01-80-C2-00-00-4F) then
 - o The frame is not propagated through an RBridge although some special processing may be done at the port as specified in [RFC6325] and the frame may be dispatched to Layer 2 processing at the port if certain protocols are supported by that port (examples: Link Aggregation Protocol, Link Layer Discovery Protocol).
- o If the DA is some other multicast value
 - o Insert TRILL-HDR and construct (TRILL-HDR, iDA, iSA, iVL, M3)
 - o Pass the (TRILL-HDR, iDA, iSA, iVL, M3) to RBridge Layer

4.3.4. Transmit Processing of Multicast packets

The following ignores the case of transmitting TRILL IS-IS packets.

- o Receive indication (TRILL-HDR, iDA, iSA, iVL, M3) from RBridge layer.
- o If TRILL-HDR multicast flag set and TRILL-HDR Alert flag set and OAM EtherType present then:
 - o (DA, SA, V, M2) by inserting TRILL Outer.MacDA of All-RBridges, Outer.MacSA, Outer.VLAN and TRILL EtherType. M2 here is (EtherType TRILL, TRILL-HDR, iDA, iSA, iVL, M)
 - NOTE: Second copy of native format is not made.
- o Else If TRILL-HDR multicast flag set and Alert flag not set
 - o If the port is appointed Forwarder for iVL and the port is not configured as a trunk port or a p2p port, Strip TRILL-HDR, iSA, iDA, iVL and construct (DA, SA, V, M2) for native format.
 - o Make a second copy (DA, SA, V, M2) by inserting TRILL Outer.MacDA, Outer.MacSA, Outer.VLAN and TRILL

EtherType. M2 here is (EtherType TRILL, TRILL-HDR, iDA, iSA, iVL, M)

- o Pass the indication (DA, SA, V, M2) to End Station VLAN processing layer.

4.4. TRILL OAM Layer Processing

TRILL OAM Processing Layer is located between the TRILL Encapsulation / De-capsulation layer and RBridge Layer. It performs the following: 1. Identification of OAM frames that need local processing and 2. performs OAM processing or redirect to the CPU for OAM processing.

- o Receive indication (TRILL-HDR, iDA, iSA, iVL, M3) from RBridge layer. M3 is the payload after inner VLAN iVL.
- o If the TRILL Multicast Flag is set and TRILL Alert Flag is set and TRILL OAM EtherType is present then
 - o If MEP or MIP is configured on the Inner.VLAN of the packet then
 - . discard packets that have MD-LEVEL Less than that of the MEP or packets that do not have MD-LEVEL present (e.g., due to packet truncation).
 - . If MD-LEVEL matches MD-LEVEL of the MEP then
 - . Re-direct to OAM Processing (Do not forward further)
 - . If MD-LEVEL matches MD-LEVEL of MIP then
 - . Make a Copy for OAM processing and continue
 - . If MD-LEVEL matches MD-LEVEL of MEP then
 - . Redirect the OAM packet to OAM processing and do not forward along or forward as a native packet.
 - o Else if TRILL Alert Flag is set and TRILL OAM EtherType is present then
 - o If MEP or MIP is configured on the Inner.VLAN of the packet then
 - . discard packets that have MD-LEVEL not present or MD-LEVEL is Less than that of the MEP.
 - . If MD-LEVEL matches MD-LEVEL of the MEP then
 - . Re-direct to OAM Processing (Do not forward further)
 - . If MD-LEVEL matches MD-LEVEL of MIP then
 - . Make a Copy for OAM processing and continue

- o Else // Non OAM l Packet
 - o Continue
- o Pass the indication (DA, SA, V, M2) to End Station VLAN processing layer.

NOTE: In the Receive path, processing above compares against Down MEP and MIP Half functions. In the transmit processing it compares against Up MEP and MIP Half functions.

Appointed Forwarder is a function the TRILL Encap/De-Cap layer performs. The TRILL Encap/De-cap Layer is responsible for prevention of leaking of OAM packets as native frames.

5. Maintenance Associations (MA) in TRILL

[8021Q] defines a maintenance association as a logical relationship between a group of nodes. Each Maintenance Association (MA) is identified with a unique MAID of 48 bytes [8021Q]. CCM and other related OAM functions operate within the scope of an MA. The definition of MA is technology independent. Similarly it is encoded within the OAM message, not in the technology dependent portion of the packet. Hence the MAID as defined in [8021Q] can be utilized for TRILL OAM, without modifications. This also allows us to utilize CCM and LBM messages defined in [8021Q], as is.

In TRILL, an MA may contain two or more RBridges (MEPs). For unicast, it is likely that the MA contains exactly two MEPs that are the two end-points of the flow. For multicast, the MA may contain two or more MEPs.

For TRILL, in addition to all of the standard [8021Q] CFM MIB definitions, each MEP's MIB contains one or more flow entropy definitions corresponding to the set of flows that the MEP monitors.

[8021Q] CFM MIB is augmented to add the TRILL specific information. Figure 5, below depicts the augmentation of the CFM MIB to add the TRILL specific Flow Entropy.

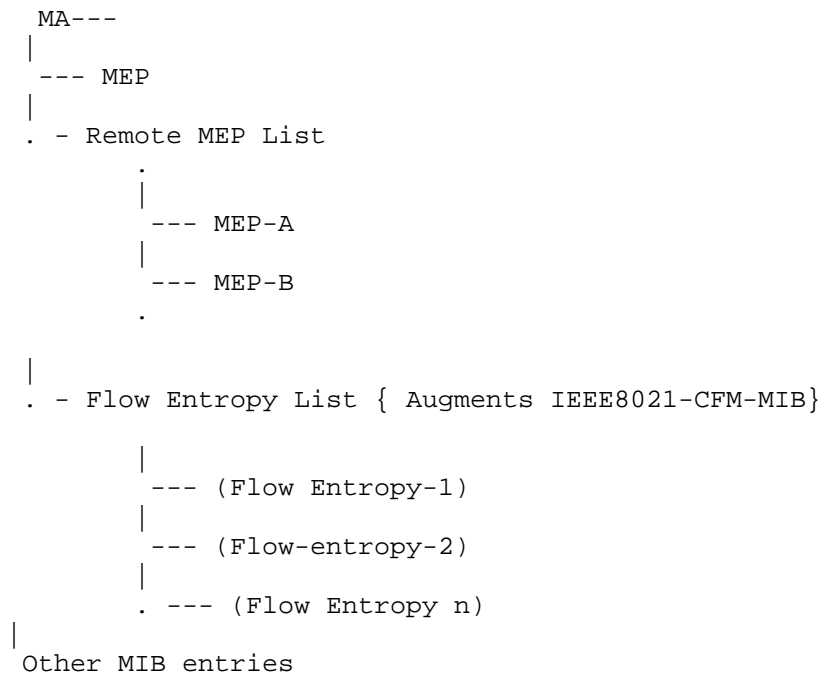


Figure 5 Correlation of TRILL augmented MIB

The detailed TRILL OAM MIB will be specified in a separate document [TRILLOAMMIB].

6. MEP Addressing

In IEEE CFM [8021Q], OAM messages address the target MEP by utilizing a unique MAC address. In TRILL a MEP is addressed by combination of the egress RBridge nickname and the Inner VLAN/FGL.

At the MEP, OAM packets go through a hierarchy of op-code de-multiplexers. The op-code de-multiplexers channel the incoming OAM packets to the appropriate message processor (e.g. LBM) The reader may refer to Figure 6 below for a visual depiction of these different de-multiplexers.

1. Identify the packets that need OAM processing at the Local RBridge as specified in Section 4.
 - a. Identify the MEP that is associated with the Inner.VLAN.
2. The MEP first validates the MD-LEVEL and then
 - a. Redirect to MD-LEVEL De-multiplexer
3. MD-LEVEL de-multiplexer compares the MD-Level of the packet against the MD level of the local MEPs of a given MD-Level on the port (Note: there can be more than one MEP at the same MD-Level but belonging to different MAs)
 - a. If the packet MD-LEVEL is equal to the configured MD-LEVEL of the MEP, then pass to the Opcode de-multiplexer
 - b. If the packet MD-LEVEL is less than the configured MD-LEVEL of the MEP, discard the packet
 - c. If the packer MD-LEVEL is greater than the configured MD-LEVEL of the MEP, then pass on to the next higher MD-LEVEL de-multiplexer, if available. Otherwise, if no such higher MD-LEVEL de-multiplexer exists, then forward the packet as normal data.
4. Opcode De-multiplexer compares the opcode in the packet with supported opcodes
 - a. If Op-code is CCM, LBM, LBR, PTM, PTR, MTVM, MTVR, then pass on to the correct Processor
 - b. If Op-code is Unknown, then discard.

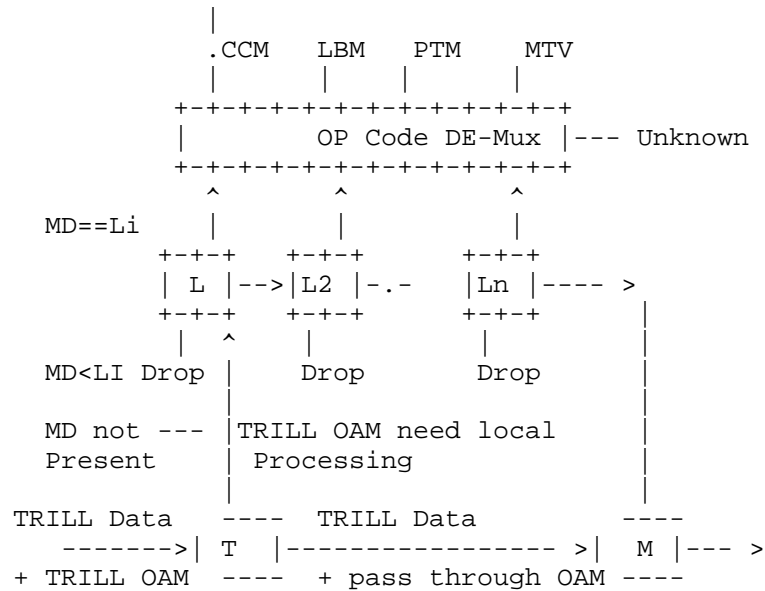


Figure 6 OAM De-Multiplexers at MEP for active SAP

T : Denotes Tap, that identifies OAM frames that need local processing. These are the packets with Alert flag set and OAM EtherType is present after the flow entropy of the packet

M : Is the post processing merge, merges data and OAM messages that are passed through. Additionally, the Merge component ensures, as explained earlier, that OAM packets are not forwarded out as native frames.

L : Denotes MD-Level processing. Packets with MD-Level less than the Level will be dropped. Packets with equal MD-Level are passed on to the opcode de-multiplexer. Others are passed on to the next level MD processors or eventually to the merge point (M).

NOTE: LBM, MTV and PT are not subject to MA de-multiplexers. These packets do not have an MA encoded in the packet. Adequate response can be generated to these packets, without loss of functionality, by any of the MEPs present on that interface or an entity within the RBridge.

6.1. Use of MIP in TRILL

Maintenance Intermediate Points (MIP) are mainly used for fault isolation. Link Trace Messages in [8021Q] utilize a well-known multicast MAC address and MIPs generate responses to Link Trace messages. Response to Link Trace messages or lack thereof can be used for fault isolation in TRILL.

As explained in section 10. , a hop-count expiry approach will be utilized for fault isolation and path tracing. The approach is very similar to the well-known IP trace-route approach. Hence, explicit addressing of MIPs is not required for the purpose of fault isolation.

Any given RBridge can have multiple MIPs located within an interface. As such, a mechanism is required to identify which MIP should respond to an incoming OAM message.

A similar approach to that presented above for MEPs can be used for MIP processing. It is important to note that "M", the merge block of a MIP, does not prevent OAM packets leaking out as native frames. On edge interfaces, MEPs MUST be configured to prevent the leaking of TRILL OAM packets out of the TRILL Campus.

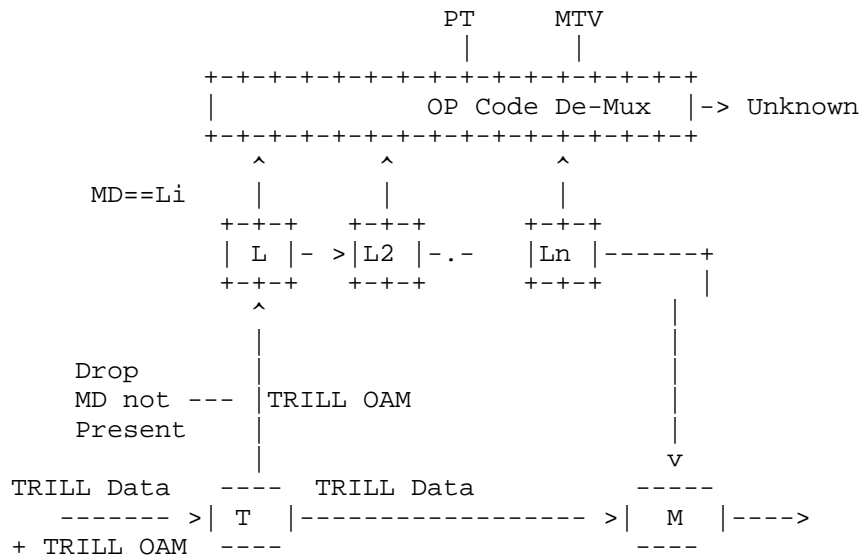


Figure 7 OAM De-Multiplexers at MIP for active SAP

T: TAP processing for MIP. All packets with OAM flag set are captured.

L : MD Level Processing, Packet with matching MD Level are "copied" to the Opcode de-multiplexer and original packet is passed on to the next MD level processor. Other packets are simply passed on to the next MD level processor, without copying to the OP code de-multiplexer.

M : Merge processor, merge OAM packets to be forwarded along with the data flow.

Packets that carry Path Trace Message (PTM) or Multi-destination Tree Verification (MTV) OpCodes are passed on to the respective processors.

Packets with unknown OpCodes are counted and discarded.

7. Continuity Check Message (CCM)

CCMs are used to monitor connectivity and configuration errors. [8021Q] monitors connectivity by listening to periodic CCM messages received from its remote MEP partners in the MA. An [8021Q] MEP identifies cross-connect errors by comparing the MAID in the received CCM message with the MEP's local MAID. The MAID [8021Q] is a 48-byte field that is technology independent. Similarly, the MEPID is a 2-byte field that is independent of the technology. Given this generic definition of CCM fields, CCM as defined in [8021Q] can be utilized in TRILL with no changes. TRILL specific information may be carried in CCMs when encoded using TRILL specific TLVs or sub-TLVs. This is possible since CCMs may carry optional TLVs.

Unlike classical Ethernet environments, TRILL contains multipath forwarding. The path taken by a packet depends on the payload of the packet. The Maintenance Association identifies the interested end-points (MEPs) of a given monitored path. For unicast there are only two MEPs per MA. For multicast there can be two or more MEPs in the MA. The entropy values of the monitored flows are defined within the MA. CCM transmit logic will utilize these flow entropy values when constructing the CCM packets. Please see section 12. below for the theory of operation of CCM.

The MIB of [8021Q] is augmented with the definition of flow-entropy. Please see [TRILLOAMMIB] for definition of these and other TRILL related OAM MIB definitions. The below Figure depicts the correlation between MA, CCM and the flow-entropy.

```

    MA---
    |
    --- MEP
    |
    . - Remote MEP List
        .
        |
        --- MEP-A
        |
        --- MEP-B
        .

    |
    . - Flow Entropy List {Augments IEEE8021-CFM-MIB}
        |
        --- (Flow Entropy-1)
        |
        --- (Flow-entropy-2)
        |
        . --- (Flow Entropy n)

    |
    . - CCM
        |
        --- (standard 8021ag entries)
        |
        --- (hop-count) { Augments IEEE8021-CFM-MIB}
        |
        --- (Other TBD TRILL OAM specific entries)
                                {Augmented}

    |
    .
    |
    - Other MIB entries

```

Figure 8 Augmentation of CCM MIB in TRILL

In a multi-pathing environment, a Flow - by definition - is unidirectional. A question may arise as to what flow entropy should be used in the response. CCMs are unidirectional and have no explicit reply; as such, the issue of the response flow entropy does not arise. In the transmitted CCM, each MEP reports local status using the Remote Defect Indication (RDI) flag. Additionally, a MEP may raise SNMP TRAPS [TRILLOAMMIB] as Alarms when a connectivity failure occurs.

8. TRILL OAM Message Channel

The TRILL OAM Message Channel can be divided into two parts: TRILL OAM Message header and TRILL OAM Message TLVs. Every OAM Message MUST contain a single TRILL OAM message header and a set of one or more specified OAM Message TLVs.

8.1. TRILL OAM Message header

As discussed earlier, a common messaging framework between [8021Q], TRILL, and other similar standards such as Y.1731 is accomplished by re-using the OAM message header defined in [8021Q].

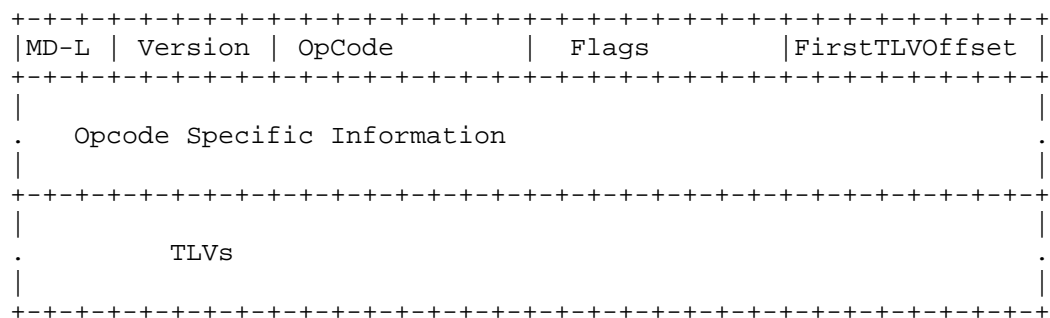


Figure 9 OAM Message Format

- o MD-L: Maintenance Domain Level (3 bits). Identifies the maintenance domain level. For TRILL, in general, this field is set to a single value across the TRILL campus. When using TRILL base mode as specified in Appendix B, MD-L is set to 3. However, extension of TRILL, for example to support multilevel, may create different MD-LEVELs and MD-L field must be appropriately set in those scenarios. (Please refer to [8021Q] for the definition of MD-Level)
- o Version: Indicates the version (5 bits) as specified in [8021Q]. This document does not require changing the Version defined in [8021Q].
- o Flags: Includes operational flags (1 byte). The definition of flags is OpCode-specific and is covered in the applicable sections.

- o FirstTLVOffset: Defines the location of the first TLV, in bytes, starting from the end of the FirstTLVOffset field (1 byte). (Refer to [8021Q] for the definition of the FirstTLVOffset.)

MD-L, Version, Opcode, Flags and FirstTLVOffset fields collectively are referred to as the OAM Message Header.

The Opcode specific information section of the OAM Message may contain Session Identification number, time-stamp, etc.

8.2. TRILL Specific OAM Opcodes

The following TRILL specific CFM Opcodes are defined. Each of the Opcodes indicates a separate type of TRILL OAM message. Details of the messages are presented in the related sections.

TRILL OAM Message Opcodes:

TBD1: Path Trace Reply
 TBD2: Path Trace Message
 TBD3: Multicast Tree Verification Reply
 TBD4: Multicast Tree Verification Message

Loopback and CCM Messages reuse the opcodes defined by [8021Q]

8.3. Format of TRILL OAM TLV

The same CFM TLV format as defined in [8021Q] is used for TRILL OAM. The following figure depicts the general format of a TRILL OAM TLV:

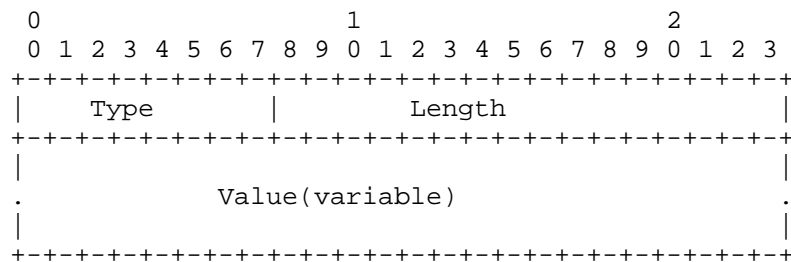


Figure 10 TRILL OAM TLV

Type (1 octet): Specifies the Type of the TLV (see sections 8.4. for TLV types).

Length (2 octets): Specifies the length of the 'Value' field in octets. Length of the 'Value' field can be either zero or more octets.

Value (variable): The length and the content of this field depend on the type of the TLV. Please refer to applicable TLV definitions for the details.

Semantics and usage of Type values allocated for TRILL OAM purpose are defined by this document and other future related documents.

8.4. TRILL OAM TLVs

TRILL related TLVs are defined in this section. [8021Q] defined TLVs are reused, where applicable.

8.4.1. Common TLVs between CFM and TRILL

The following TLVs are defined in [8021Q]. We re-use them where applicable. The format and semantics of the TLVs are as defined in [8021Q].

Type	Name of TLV in [8021Q]
0	End TLV
1	Sender ID TLV
2	Port Status TLV
3	Data TLV
4	Interface Status TLV
5	Reply Ingress TLV
6	Reply Egress TLV
7	LTM Egress Identifier TLV
8	LTR Egress Identifier TLV
9-30	Reserved
31	Organization Specific TLV

8.4.2. TRILL OAM Specific TLVs

Listed below is a summary of TRILL OAM TLVs and their corresponding codes. Format and semantics of TRILL OAM TLVs are defined in subsequent sections.

Type	TLV Name
-----	-----
TBDa	TRILL OAM Application Identifier
TBDb	Out of Band IP Address
TBDc	Original Payload
TBDd	Diagnostic VLAN
TBDe	RBridge scope
TBDf	Previous RBridge Nickname
TBDg	TRILL Next Hop RBridge List (ECMP)
TBDh	Multicast Receiver Availability
TBDi	Flow Identifier
TBDj	Reflector Entropy

8.4.3. TRILL OAM Application Identifier TLV

TRILL OAM Application Identifier TLV carries TRILL OAM application specific information. The TRILL OAM Application Identifier TLV MUST always be present and MUST be the first TLV in TRILL OAM messages. Messages that do not include the TRILL OAM Application Identifier TLV as the first TLV MUST be discarded by TRILL MP.

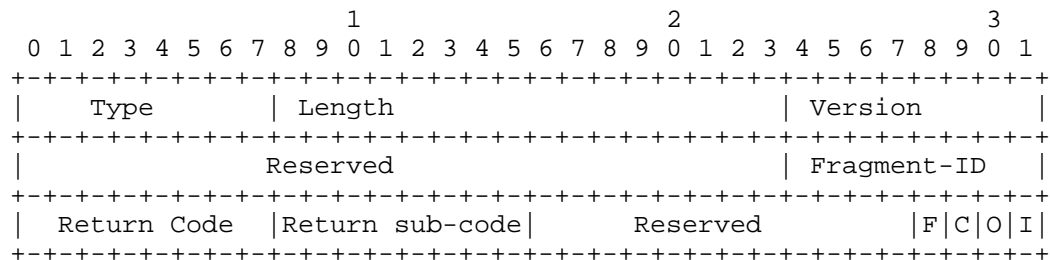


Figure 11 TRILL OAM Application Identifier TLV

Type (1 octet) = TBDa indicate that this is the TRILL OAM Application Identifier TLV

Length (2 octets) = 6

TRILL OAM Version (1 Octet), currently set to zero. Indicates the TRILL OAM version. TRILL OAM version can be different than the [8021Q] version.

Fragment-ID (1 octet): Indicates the fragment number of the current message. This applies only to the reply messages. F flag defined below MUST be set with the final message whether it is the last fragment of the fragmented message or only message of the reply. Section 13. below provide more details.

Return Code (1 Octet): Set to zero on requests. Set to an appropriate value in response messages.

Return sub-code (1 Octet): Return sub-code is set to zero on transmission of request message. Return sub-code identifies categories within a specific Return code. Return sub-code MUST be interpreted within a Return code.

Reserved: set to zero on transmission and ignored on reception.

F (1 bit): Final flag, when set, indicates this is the last response.

C (1 bit): Label error (VLAN/Label mapping error), if set indicates that the label (VLAN/FGL) in the flow entropy is different than the label included in the diagnostic TLV. This field is ignored in request messages and MUST only be interpreted in response messages.

O (1 bit): If set, indicates, OAM out-of-band response requested.

I (1 bit): If set, indicates, OAM in-band response requested.

NOTE: When both O and I bits are set to zero, indicates that no response is required (silent mode). User MAY specify both O and I or one of them or none. When both O and I bits are set response is sent both in-band and out-of-band.

8.4.4. Out Of Band Reply Address TLV

Out of Band Reply Address TLV specifies the address to which an out of band OAM reply message MUST be sent. When O bit in the TRILL Version TLV is not set, Out of Band Reply Address TLV is ignored.

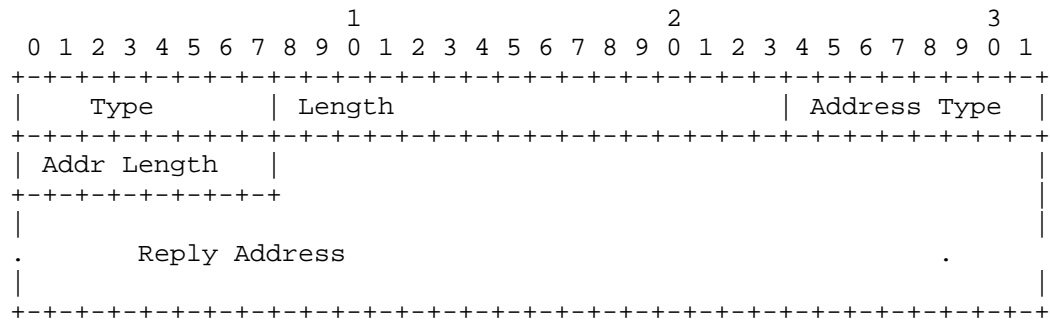


Figure 12 Out of Band IP Address TLV

Type (1 octet) = TBDb

Length (2 octets) = Variable. Minimum length is 2.

Address Type (1 Octet): 0 - IPv4. 1 - IPv6. 2 - TRILL RBridge nickname. All other values reserved.

Addr Length (1 Octet). 4 - IPv4. 16 - IPv6, 2 - TRILL RBridge nickname.

Reply Address (variable): Address where the reply needed to be sent. Length depends on the address specification.

8.4.5. Diagnostics Label TLV

Diagnostic label specifies the data label (VLAN or FGL) in which the OAM messages are generated. Receiving RBridge MUST compare the data label of the Flow entropy to the data label specified in the Diagnostic Label TLV. Label Error Flag in the response (TRILL OAM Message Version TLV) MUST be set when the two VLANs do not match.

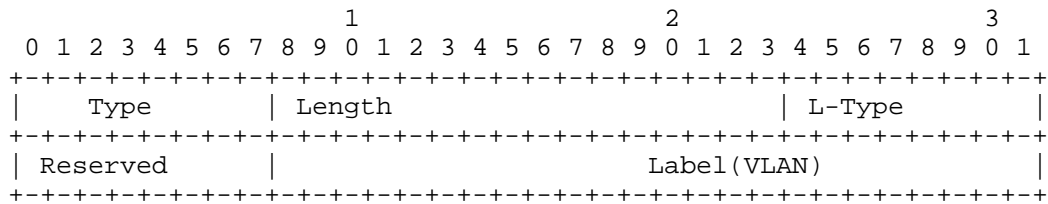


Figure 13 Diagnostic VLAN TLV

Type (1 octet) = TBDC indicates that this is the TRILL Diagnostic VLAN TLV

Length (2 octets) = 5

L-Type (Label type, 1 octet)

0- indicate 802.1Q 12 bit VLAN.

1 - indicate TRILL 24 bit fine grain label

Label (24 bits): Either 12 bit VLAN or 24 bit fine grain label.

RBridges do not perform Label error checking when the Label TLV is not included in the OAM message. In certain deployments intermediate devices may perform label translation. In such scenarios, originator should not include the diagnostic Label TLV in OAM messages. Inclusion of diagnostic TLV will generate unwanted label error notifications.

8.4.6. Original Data Payload TLV

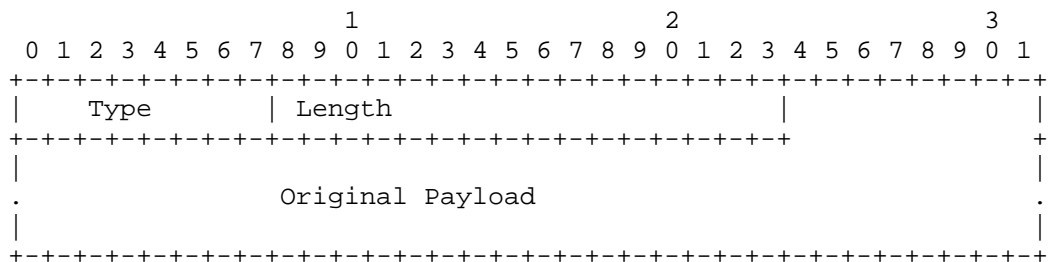


Figure 14 Original Data Payload TLV

Type (1 Octet) = TBDd

Length (2 octets) = variable

Original Payload: The original TRILL Header and Entropy. Used in constructing replies to the Loopback Message (see Section 9) and the Path Trace Message (see Section 10).

8.4.7. RBridge scope TLV

RBridge scope TLV identifies nicknames of RBridges from which a response is required. The RBridge scope TLV is only applicable to Multicast Tree Verification messages. This TLV SHOULD NOT be included in other messages. Receiving RBridges MUST ignore this TLV on messages other than Multicast Verification Message.

Each TLV can contain up to 255 nicknames of in-scope RBridges. A Multicast Verification Message may contain multiple "RBridge scope TLVs", in the event that more than 255 in scope RBridges need to be specified.

Absence of the "RBridge scope TLV" indicates that a response is needed from all the RBridges. Please see section 11. for details.

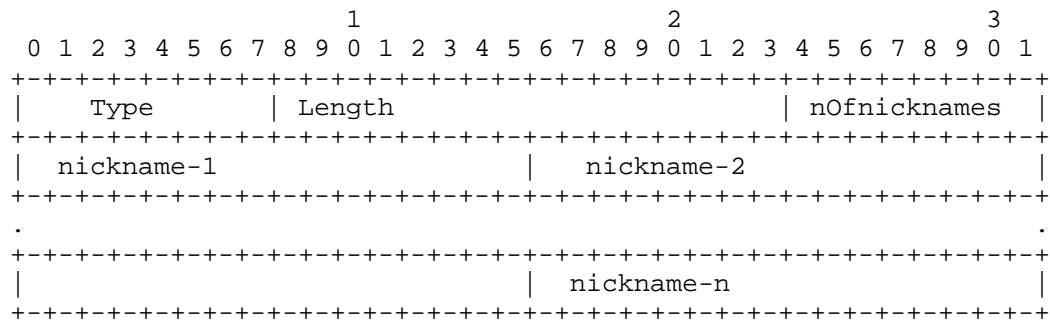


Figure 15 RBridge Scope TLV

Type (1 octet) = TBDe indicates that this is the "RBridge scope TLV"

Length (2 octets) = variable. Minimum value is 2.

Nickname (2 octets) = 16 bit RBridge nickname.

8.4.8. Previous RBridge nickname TLV

"Previous RBridge nickname TLV" identifies the nickname or nicknames of the upstream RBridge. [RFC6325] allows a given RBridge to hold multiple nicknames.

"Upstream RBridge nickname TLV" is an optional TLV. Multiple instances of this TLV MAY be included when an upstream RBridge is represented by more than 255 nicknames (highly unlikely).

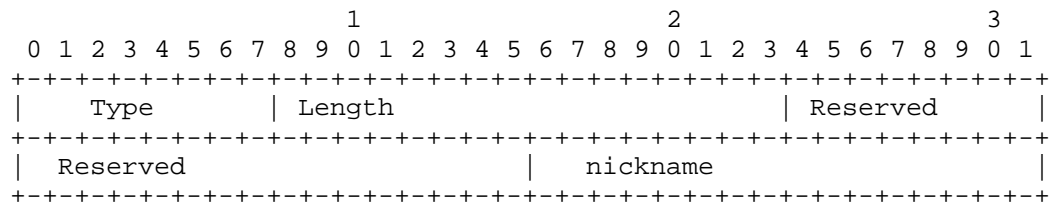


Figure 16 Previous RBridge nickname TLV

Type (1 octet) = TBDf indicates that this is the "Upstream RBridge nickname"

Length (2 octets) = 4.

Nickname (2 octets) = 16 bit RBridge nickname.

8.4.9. Next Hop RBridge List TLV

"Next Hop RBridge List TLV" identifies the nickname or nicknames of the downstream next hop RBridges. [RFC6325] allows a given RBridge to have multiple Equal Cost Paths to a specified destination. Each next hop RBridge is represented by one of its nicknames.

"Next Hop RBridge List TLV" is an optional TLV. Multiple instances of this TLV MAY be included when there are more than 255 Equal Cost Paths to the destination.

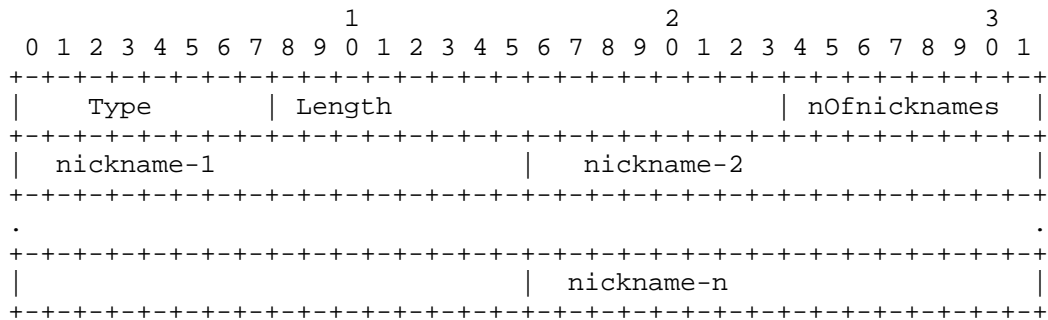


Figure 17 Next Hop RBridge List TLV

Type (1 octet) = TBDg indicates that this is the "Next nickname"

Length (2 octets) = variable. Minimum value is 2.

Nickname (2 octets) = 16 bit RBridge nickname.

8.4.10. Multicast Receiver Port count TLV

"Multicast Receiver Port Count TLV" identifies the number of ports interested in receiving the specified multicast stream within the responding RBridge on the label (VLAN or FGL) specified by the Diagnostic Label TLV.

Multicast Receiver Port count is an Optional TLV.

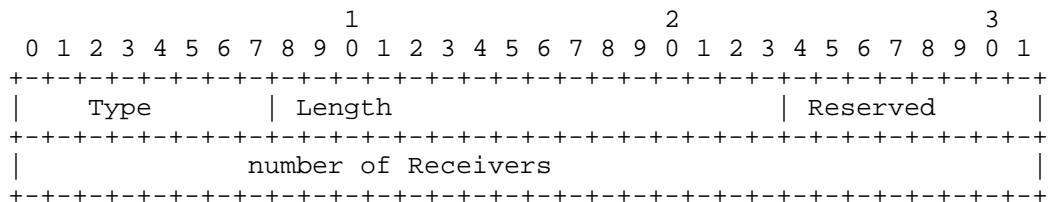


Figure 18 Multicast Receiver Availability TLV

Type (1 octet) = TBDh indicates that this is the "Multicast Availability TLV"

Length (2 octets) = 5.

Number of Receivers (4 octets) = Indicates the number of Multicast receivers available on the responding RBridge on the label specified by the diagnostic label.

8.4.11. Flow Identifier (flow-id) TLV

Flow Identifier (flow-id) uniquely identifies a specific flow. The flow-id value is unique per MEP and needs to be interpreted as such.

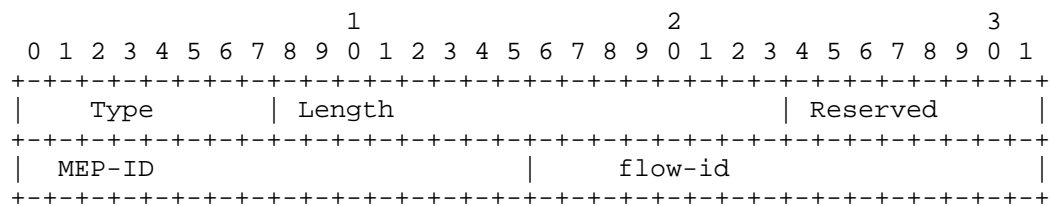


Figure 19 Flow Identifier TLV

Type (1 octet) = TBDi

Length (2 octets) = 5.

Reserved (1 octet) set to 0 on transmission and ignored on reception.

MEP-ID (2 octets) = MEP-ID of the originator [8021Q].

Flow-id (2 octets) = uniquely identifies the flow per MEP. Different MEPs may allocate the same flow-id value. The {MEP-ID, flow-id} pair is globally unique.

Inclusion of the MEP-ID in the flow-id TLV allows the inclusion of a MEP-ID for messages that do not contain a MEP-ID in their OAM header. Applications may use MEP-ID information for different types of troubleshooting.

8.4.12. Reflector Entropy TLV

Reflector Entropy TLV is an optional TLV. This TLV, when present, tells the responder to utilize the Reflector Entropy specified within the TLV as the flow-entropy of the response message.

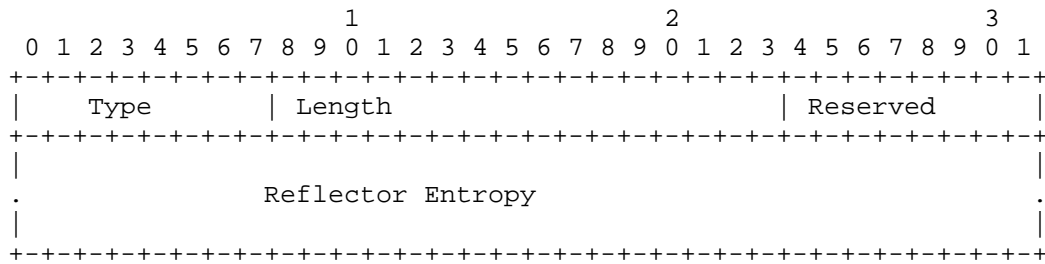


Figure 20 Reflector Entropy TLV

Type (1 octet) =TBDj Reflector Entropy TLV.

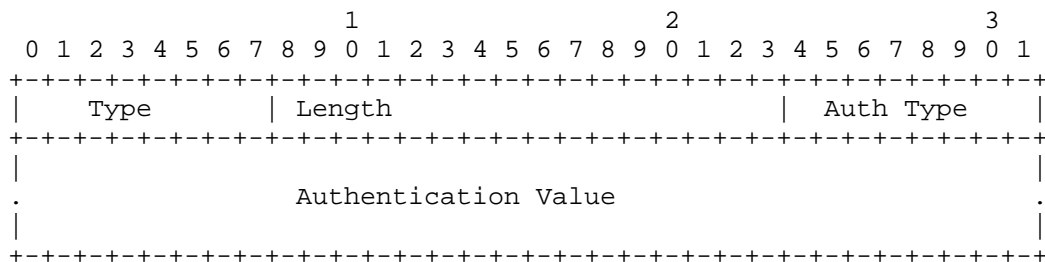
Length (1 octet) =97.

Reserved (1 octet) = set to zero on transmission and ignored by the recipient.

Reflector Entropy (96-octet) = Flow Entropy to be used by the responder. May be padded with zero if the desired flow entropy is less than 96 octets.

8.4.13. OAM Authentication TLV

The Authentication TLV is an optional TLV that can appear in any OAM Message or Reply in TRILL.



Type (1 octet) =TBDk Authentication TLV.

Length (1 octet) = variable length

The Auth Type and following Authentication Value are the same as the Auth Type and following value for the [IS-IS] Authentication TLV. It is RECOMMENDED that Auth Type 3 be used, in which case the process is generally as specified in [RFC5310] using the Key ID space as TRILL IS-IS. The area covered by the Authentication TLV is from the beginning of the TRILL Header to the end of the TRILL OAM Message Channel - the Link Header and Trailer are not included. The TRILL Header Alert and Reserved bit and Hop Count are treated as if zero for the purposes of computing and verifying the Authentication Data.

An RBridge supporting OAM authentication can be configured to either (1) ignore received OAM Authentication TLVs and not send them, (2) ignore received OAM Authentication TLVs but include them in all OAM packets sent, or (3) to include Authentication TLVs in all OAM messages sent and enforce authentication of OAM messages received. When an RBridge is enforcing authentication, it discards any OAM message subject to OAM processing that does not contain an Authentication TLV or if the Authentication TLV does not verify.

9. Loopback Message

9.1. Loopback OAM Message format

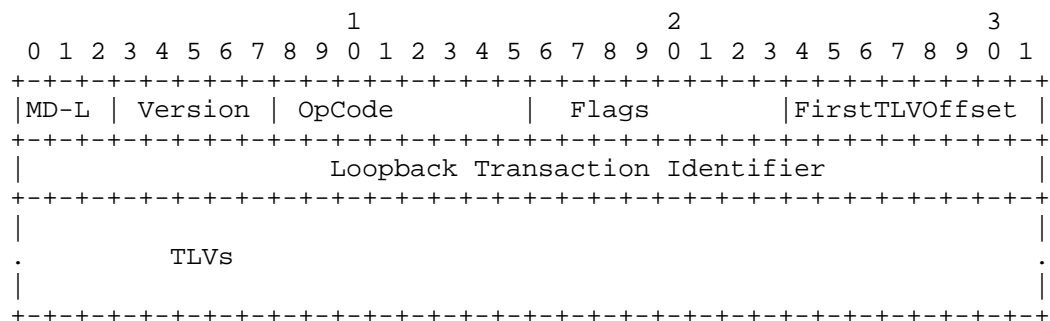


Figure 21 Loopback OAM Message Format

The above figure depicts the format of the Loopback Request and response messages as defined in [8021Q]. The Opcode for Loopback Message is set to 3 and the Opcode for the Reply Message is set to 2 [8021Q]. The Session Identification Number is a 32-bit

integer that allows the requesting RBridge to uniquely identify the corresponding session. Responding RBridges, without modification, MUST echo the received "Loopback Transaction Identifier" number.

9.2. Theory of Operation

9.2.1. Actions by Originator RBridge

The originator RBridge takes the following actions:

Identifies the destination RBridge nickname based on user specification or based on the specified destination MAC or IP address.

Constructs the flow entropy based on user specified parameters or implementation specific default parameters.

Constructs the TRILL OAM header: sets the opcode to Loopback message type (3)[8021Q]. Assign applicable Loopback Transaction Identifier number for the request.

The TRILL OAM Version TLV MUST be included and with the flags set to applicable values.

Include following OAM TLVs, where applicable

- o Out-of-band Reply address TLV
- o Diagnostic Label TLV
- o Sender ID TLV

Specify the Hop count of the TRILL data frame per user specification or utilize an applicable Hop count value.

Dispatch the OAM frame for transmission.

RBridges may continue to retransmit the request at periodic intervals, until a response is received or the re-transmission count expires. At each transmission Session Identification number MUST be incremented.

9.2.2. Intermediate RBridge

Intermediate RBridges forward the frame as a normal data frame and no special handling is required.

9.2.3. Destination RBridge

If the Loopback message is addressed to the local RBridge and satisfies the OAM identification criteria specified in section 3.1. then, the RBridge data plane forwards the message to the CPU for further processing.

The TRILL OAM application layer further validates the received OAM frame by checking for the presence of OAM-Ethertype at the end of the flow entropy. Frames that do not contain OAM-Ethertype at the end of the flow entropy MUST be discarded.

Construction of the TRILL OAM response:

TRILL OAM application encodes the received TRILL header and flow entropy in the Original payload TLV and includes it in the OAM message.

Set the Return Code to (1) "Reply" and Return sub code to zero (0) "Valid Response". Update the TRILL OAM opcode to 2 (Loopback Message Reply)

Optionally, if the VLAN/FGL identifier value of the received flow entropy differs from the value specified in the diagnostic Label, set the Label Error Flag on TRILL OAM Application Identifier TLV.

Include the sender ID TLV (1)

If in-band response was requested, dispatch the frame to the TRILL data plane with request-originator RBridge nickname as the egress RBridge nickname.

If out-of-band response was requested, dispatch the frame to the IP forwarding process.

10. Path Trace Message

The primary use of the Path Trace Message is for fault isolation. It may also be used for plotting the path taken from a given RBridge to another RBridge.

[8021Q] accomplishes the objectives of the TRILL Path Trace Message using Link Trace Messages. Link Trace Messages utilize a well-known multicast MAC address. This works for [8021Q], because for 802.1 both the unicast and multicast paths are congruent.

However, in TRILL multicast and unicast are not congruent. Hence, TRILL OAM uses a new message format: the Path Trace message.

The Path Trace Message has the same format as Loopback Message. The Opcode for Path Trace Reply is TBD1 and for Path Trace Message is TBD2.

Operation of the Path Trace message is identical to the Loopback message except that it is first transmitted with a TRILL Header Hop count field value of 1. The sending RBridge expects a Time Expiry Return-Code from the next hop or a successful response. If a Time Expiry Return-code is received as the response, the originator RBridge records the information received from intermediate node that generated the Time Expiry message and resends the message by incrementing the previous Hop count value by 1. This process is continued until, a response is received from the destination RBridge or Path Trace process timeout occur or Hop count reaches a configured maximum value.

10.1. Theory of Operation

10.1.1. Action by Originator RBridge

Identify the destination RBridge based on user specification or based on location of the specified MAC address.

Construct the flow entropy based on user specified parameters or implementation specific default parameters.

Construct the TRILL OAM header: Set the opcode to Path Trace Request message type (TBD2). Assign an applicable Session Identification number for the request. Return-code and sub-code MUST be set to zero.

The TRILL OAM Application Identifier TLV MUST be included and set the flags to applicable values.

Include following OAM TLVs, where applicable

- o Out-of-band IP address TLV
- o Diagnostic Label TLV
- o Include the Sender ID TLV

Specify the Hop count of the TRILL data frame as 1 for the first request.

Dispatch the OAM frame to the TRILL data plane for transmission.

An RBridge may continue to retransmit the request at periodic intervals, until a response is received or the re-transmission count expires. At each new re-transmission, the Session Identification number MUST be incremented. Additionally, for responses received from intermediate RBridges, the RBridge nickname and interface information MUST be recorded.

10.1.2. Intermediate RBridge

Path Trace Messages transit through Intermediate RBridges transparently, unless Hop-count has expired.

TRILL OAM application layer further validates the received OAM frame by examining the presence of TRILL Alert Flag and OAM-Ethertype at the end of the flow entropy and by examining the MD Level. Frames that do not contain OAM-Ethertype at the end of the flow entropy MUST be discarded.

Construction of the TRILL OAM response:

TRILL OAM application encodes the received TRILL header and flow entropy in the Original payload TLV and include it in the OAM message.

Set the Return Code to (1) "Reply" and Return sub code to zero (0) "Valid Response". Update the TRILL OAM opcode to TBD1 (Path Trace Message Reply).

If the VLAN/FGL identifier value of the received flow entropy differs from the value specified in the diagnostic Label, set the Label Error Flag on TRILL OAM Application Identifier TLV.

Include following TLVs

Upstream RBridge nickname TLV (69)

Reply Ingress TLV (5)

Reply Egress TLV (6)

Interface Status TLV (4)

TRILL Next Hop RBridge (Repeat for each ECMP) (70)

Sender ID TLV (1)

If Label error detected, set C flag (Label error detected) in the version.

If in-band response was requested, dispatch the frame to the TRILL data plane with request-originator RBridge nickname as the egress RBridge nickname.

If out-of-band response was requested, dispatch the frame to the standard IP forwarding process.

10.1.3. Destination RBridge

Processing is identical to section 10.1.2. With the exception that TRILL OAM Opcode is set to Path Trace Reply (TBD1).

11. Multi-Destination Tree Verification (MTV) Message

Multi-Destination Tree Verification messages allow verifying TRILL distribution tree integrity and pruning. TRILL VLAN/FGL and multicast pruning are described in [RFC6325] [RFCclcorrect] and [RFCfgl]. Multi-destination tree verification and Multicast group verification messages are designed to detect pruning defects. Additionally, these tools can be used for plotting a given multicast tree within the TRILL campus.

Multi-Destination tree verification OAM frames are copied to the CPU of every intermediate RBridge that is part of the distribution tree being verified. The originator of the Multi-destination Tree verification message specifies the scope of RBridges from which a response is required. Only the RBridges listed in the scope field respond to the request. Other RBridges silently discard the request. Inclusion of the scope parameter is required to prevent receiving an excessive number of responses. The typical scenario of distribution tree verification or group verification, involves verifying multicast connectivity to a selected set of end-nodes as opposed to the entire network. Availability of the scope facilitates narrowing down the focus to only the RBridges of interest.

Implementations MAY choose to rate-limit CPU bound multicast traffic. As a result of rate-limiting or due to other congestion conditions, MTV messages may be discarded from time to time by the intermediate RBridges and the requester may be required to

retransmit the request. Implementations SHOULD narrow the embedded scope of retransmission request only to R Bridges that have failed to respond.

11.1. Multi-Destination Tree Verification (MTV) OAM Message Format

Format of MTV OAM Message format is identical to that of Loopback Message format defined in section 9. with the exception that the Loopback Transaction Identifier, in section 9.1. , is replaced with the Session Identifier and the Op-Codes used is.

11.2. Theory of Operation

11.2.1. Actions by Originator R Bridge

The user is required at a minimum to specify either the distribution trees that need to be verified, or the Multicast MAC address and VLAN/FGL, or VLAN/FGL and Multicast destination IP address. Alternatively, for more specific multicast flow verification, the user MAY specify more information e.g. source MAC address, VLAN/FGL, Destination and Source IP addresses. Implementations, at a minimum, must allow the user to specify a choice of distribution trees, Destination Multicast MAC address and VLAN/FGL that needs to be verified. Although, it is not mandatory, it is highly desired to provide an option to specify the scope. It should be noted that the source MAC address and some other parameters may not be specified if the Backwards Compatibility Method of Appendix A is used to identify the OAM frames.

Default parameters MUST be used for unspecified parameters. Flow entropy is constructed based on user specified parameters and/or default parameters.

Based on user specified parameters, the originating R Bridge identifies the nickname that represents the multicast tree.

Obtain the applicable Hop count value for the selected multicast tree.

Construct TRILL OAM message header and include Session Identification number. Session Identification number facilitate the originator mapping the response to the correct request.

TRILL OAM Application Identifier TLV MUST be included.

Op-Code MUST be specified as Multicast Tree Verification Message (TBD4)

Include RBridge scope TLV (TBDe)

Optionally, include following TLV, where applicable

- o Out-of-band IP address (TBDb)
- o Diagnostic Label (TBDD)
- o Sender ID TLV (1)

Specify the Hop count of the TRILL data frame per user specification or alternatively utilize the applicable Hop count value if TRILL Hop count is not being specified by the user.

Dispatch the OAM frame to the TRILL data plane to be ingressed for transmission.

The RBridge may continue to retransmit the request at a periodic interval until either a response is received or the re-transmission count expires. At each new re-transmission, the Session Identification number MUST be incremented. At each re-transmission, the RBridge may further reduce the scope to the RBridges that it has not received a response from.

11.2.2. Receiving RBridge

Receiving RBridges identify multicast verification frames per the procedure explained in sections 3.2.

The CPU of the RBridge validates the frame and analyzes the scope RBridge list. If the RBridge scope TLV is present and the local RBridge nickname is not specified in the scope list, it will silently discard the frame. If the local RBridge is specified in the scope list OR RBridge scope TLV is absent, the receiving RBridge proceeds with further processing as defined in section 11.2.3.

11.2.3. In scope RBridges

Construction of the TRILL OAM response:

TRILL OAM application encodes the received TRILL header and flow entropy in the Original payload TLV and includes them in the OAM message.

Set the Return Code to (0) and Return sub code to zero (0).
Update the TRILL OAM opcode to TBD3 (Multicast Tree Verification Reply).

Include following TLVs:

Previous RBridge nickname TLV (TBDF)

Reply Ingress TLV (5)

Interface Status TLV (4)

TRILL Next Hop RBridge List (TBDg)

Sender ID TLV (1)

Multicast Receiver Availability TLV (TBDh)

If a Label (VLAN or FGL) cross connect error detected, set the C flag (Cross connect error detected) in the version.

If in-band response was requested, dispatch the frame to the TRILL data plane with request-originator RBridge nickname as the egress RBridge nickname.

If out-of-band response was requested, dispatch the frame to the standard IP forwarding process.

12. Application of Continuity Check Message (CCM) in TRILL

Section 7. provides an overview of CCM Messages defined in [8021Q] and how they can be used within the TRILL OAM. This section, presents the application and Theory of Operations of CCM within the TRILL OAM framework. Readers are referred to [8021Q] for CCM message format and applicable TLV definitions and usages. Only the TRILL specific aspects are explained below.

In TRILL, between any two given MEPs there can be multiple potential paths. Whereas in [8021Q], there is always a single path between any two MEPs at any given time. [RFC6905] requires solutions to have the ability to monitor continuity over one or more paths.

CCM Messages are uni-directional, such that there is no explicit response to a received CCM message. Connectivity status is indicated by setting the applicable flags (e.g. RDI) of the CCM messages transmitted by an MEP.

It is important that the solution presented in this document accomplishes the requirements specified in [RFC6905] within the framework of [8021Q] in a straightforward manner and with minimum changes. Section 8 above defines multiple flows within the CCM object, each corresponding to a flow that a given MEP wishes to monitor.

Receiving MEPs do not cross check whether a received CCM belongs to a specific flow from the originating RBridge. Any attempt to track status of individual flows may explode the amount of state information that any given RBridge has to maintain.

The obvious question arises: How does the originating RBridge know which flow or flows are at fault?

This is accomplished with a combination of the RDI flag in the CCM header, flow-id TLV, and SNMP Notifications (Traps). Section 12.1. below discuss the procedure.

12.1. CCM Error Notification

Each MEP transmits 4 CCM messages per each flow. ([8021Q] detects CCM fault when 3 consecutive CCM messages are lost). Each CCM Message has a unique sequence number and unique flow-identifier. The flow identifier is included in the OAM message via flow-id TLV.

When an MEP notices a CCM timeout from a remote MEP (MEP-A), it sets the RDI flag on the next CCM message it generates. Additionally, it logs and sends SNMP notification that contain the remote MEP Identification, flow-id and the Sequence Number of the last CCM message it received and if available, the flow-id and the Sequence Number of the first CCM message it received after the failure. Each MEP maintains a unique flow-id per each flow, hence the operator can easily identify flows that correspond to the specific flow-id.

The following example illustrates the above.

Assume there are two MEPs, MEP-A and MEP-B.

Assume there are 3 flows between MEP-A and MEP-B.

Let's assume MEP-A allocates sequence numbers as follows

Flow-1 Sequence={1,2,3,4,13,14,15,16,... } flow-id=(1)

Flow-2 Sequence={5,6,7,8,17,18,19,20,... } flow-id=(2)

Flow-3 Sequence={9,10,12,11,21,22,23,24,... } flow-id=(3)

Let's Assume Flow-2 is at fault.

MEP-B, receives CCM from MEP-A with sequence numbers 1,2,3,4, but did not receive 5,6,7,8. CCM timeout is set to 3 CCM intervals in [8021Q]. Hence MEP-B detects the error at the 8'th CCM message. At this time the sequence number of the last good CCM message MEP-B has received from MEP-A is 4 and flow-id of the last good CCM Message is (1). Hence MEP-B will generate a CCM error SNMP notification with MEP-A and Last good flow-id (1) and sequence number 4.

When MEP-A switches to flow-3 after transmitting flow-2, MEP-B will start receiving CCM messages. In the foregoing example it will be CCM message with Sequence Numbers 9,10,11,12,21 and so on. When in receipt of a new CCM message from a specific MEP, after a CCM timeout, the TRILL OAM will generate an SNMP Notification of CCM resume with remote MEP-ID and the first valid flow-id and the Sequence number after the CCM timeout. In the foregoing example, it is MEP-A, flow-id (3) and Sequence Number 9.

The remote MEP list under the CCM MIB Object is augmented to contain "Last Sequence Number", flow-id and "CCM Timeout" variables. Last Sequence Number and flow-id are updated every time a CCM is received from a remote MEP. CCM Timeout variable is set when the CCM timeout occurs and is cleared when a CCM is received.

12.2. Theory of Operation

12.2.1. Actions by Originator RBridge

Derive the flow entropy based on flow entropy specified in the CCM Management object.

Construct the TRILL CCM OAM header as specified in [8021Q].

TRILL OAM Version TLV MUST be included as the first TLV and set the flags to applicable values.

Include other TLVs specified in [8021Q]

Include the following optional TLV, where applicable

- o Sender ID TLV (1)

Specify the Hop count of the TRILL data frame per user specification or utilize an applicable Hop count value.

Dispatch the OAM frame to the TRILL data plane for transmission.

An RBridge transmits a total of 4 requests, each at CCM retransmission interval. At each transmission the Session Identification number **MUST** be incremented by one.

At the 5'th retransmission interval, flow entropy of the CCM packet is updated to the next flow entropy specified in the CCM Management Object. If current flow entropy is the last flow entropy specified, move to the first flow entropy specified and continue the process.

12.2.2. Intermediate RBridge

Intermediate RBridges forward the frame as a normal data frame and no special handling is required.

12.2.3. Destination RBridge

If the CCM Message is addressed to the local RBridge or multicast and satisfies OAM identification methods specified in sections 3.2. then the RBridge data plane forwards the message to the CPU for further processing.

The TRILL OAM application layer further validates the received OAM frame by examining the presence of OAM-Ethertype at the end of the flow entropy. Frames that do not contain OAM-Ethertype at the end of the flow entropy **MUST** be discarded.

Validate the MD-LEVEL and pass the packet to the Opcode de-multiplexer. The Opcode de-multiplexer delivers CCM packets to the CCM process.

The CCM Process performs processing specified in [8021Q].

Additionally the CCM process updates the CCM Management Object with the sequence number of the received CCM packet. Note: The last received CCM sequence number and CCM timeout are tracked per each remote MEP.

If the CCM timeout is true for the sending remote MEP, then clear the CCM timeout in the CCM Management object and generate the SNMP notification as specified above.

13. Fragmented Reply

TRILL OAM allows Fragmented reply messages. In case of Fragmented Replies, all part of the reply MUST follow the procedure defined in this section.

The same session Identification Number MUST be included in all related fragments of the same message.

The TRILL OAM Application Identifier TLV MUST be included, with fragment-ID field monotonically increasing with each fragment transmitted with the appropriate Final Flag field. The Final Flag, MUST, only be equal to one on the final fragment of the reply.

On the receiver, process MUST order the fragments based on the fragment id. Any fragments received after final fragment MUST be discarded. Messages with incomplete fragments (i.e. messages with one or missing fragments after the receipt of the fragment with the final flag set) MUST be discarded as well.

If number of fragments exceed the maximum supported fragments (255), then return code of MUST be set according to the message and return sub code MUST be set to 1 indicating fragment limit exceed.

14. Security Considerations

Forged OAM packets could cause false error or failure indications or mask actual errors or failures. For protection against forged OAM packets, the Authentication TLV (see Section 8.4.13) can be used in and OAM message in TRILL but is, of course, ineffective unless verified.

For general TRILL related security considerations, please refer to [RFC6325].

[8021Q] requires that the MEP filters or pass through OAM messages based on the MD-Level. The MD-Level is embedded deep in the OAM message. Hence, conventional methods of frame filtering may not be able to filter frames based on the MD-Level. As a

result, OAM messages that must be dropped due to MD level mismatch may leak into a TRILL domain with different MD-Level.

This leaking may not cause any functionality loss. The receiving MEP/MIP is required to validate the MD-level prior to acting on the message. Any frames received with an incorrect MD-Level need to be dropped.

Generally, a single operator manages each TRILL campus, hence there is no risk of security exposure. However, in the event of multi operator deployments, operators should be aware of possible exposure of device specific information and appropriate measures must be taken.

It is also important to note that the MPLS OAM [RFC4379] framework does not include the concept of domains and OAM filtering based on operators. It is our opinion that the lack of OAM frame filtering based on domains does not introduce significant functional deficiency or security risk.

It is possible to mandate requiring different credentials to use different OAM functions or capabilities within a specific OAM function. Implementations may consider grouping users to different security clearance levels and restricting functions and capabilities to different clearance levels. However, Exact implementation details of such a framework are outside the scope of this document.

15. IANA Considerations

IANA is requested to assign the following:

15.1. OAM Capability Flags

Assign two TRILL-VER sub-TLV Capability Flags (see Section 3.3) as follows:

Bit	Description	Reference
---	-----	-----
TBD[2]	OAM capable	[this document]
TBD[3]	Backwards compatible OAM	[this document]

15.2. CFM Code Points

IANA is requested to assign four Op-Codes from the CFM OAM IETF Op-Codes sub-registry as follows [suggested values in square brackets]:

Value =====	Assignment =====	Reference =====
TBD1[64]	Path Trace Reply	[this document]
TBD2[65]	Path Trace Message	[this document]
TBD3[66]	Multicast Tree Verification Reply	[this document]
TBD4[67]	Multicast Tree Verification Messages	[this document]

IANA is requested to assign eleven TLV Types from the CFM OAM IETF TLV Types sub-registry as follows [suggested values in square brackets]:

Value =====	Assignment =====	Reference =====
TBDa[64]	TRILL OAM Application Identifier	[this document]
TBDb[65]	Out of Band IP Address	[this document]
TBDc[66]	Original Payload	[this document]
TBDd[67]	Diagnostic VLAN	[this document]
TBDe[68]	RBridge Scope	[this document]
TBDf[69]	Previous RBridge Nickname	[this document]
TBDg[70]	TRILL Next Hop RBridge List	[this document]
TBDh[71]	Multicast Receiver Availability	[this document]
TBDi[72]	Flow Identifier	[this document]
TBDj[73]	Reflector Entropy	[this document]
TBDk[74]	Authentication	[this document]

15.3. MAC Addresses

IANA is requested to assigned a unicast and a multicast MAC address under the IANA OUI, for identification of OAM packets as discussed for the backward compatibility method (Appendix A,

Section A.2) based on the request template in Appendix C. The assigned addresses are TBDmac1 and TBDmac2.

15.4. Return codes and sub codes

Return code zero (0) is reserved for request messages.

Return code one (1) indicate reply message. Sub code zero (0) indicates valid response, sub code of one (1) indicates fragment limit exceeded Section 13.

16. References

16.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5310] Bhatia, M., "IS-IS Cryptographic Generic Cryptographic Authentication", RFC 5310, February 2009.
- [RFC6325] Perlman, R., et.al., "Routing Bridges (Rbridges): Base Protocol Specification", RFC 6325, July 2011.
- [RFCfgl] D. Eastlake, M. Zhang, P. Agarwal, R. Perlman, D. Dutt, "TRILL: Fine-Grained Labeling", draft-ietf-trill-fine-labeling, work in progress.
- [8021Q] IEEE, "Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q-2011, August, 2011.
- [ISIS] 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.

16.2. Informative References

- [RFC4379] Kompella, K. et.al, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

- [RFC6291] Andersson, L., et.al., "Guidelines for the use of the "OAM" Acronym in the IETF" RFC 6291, June 2011.
- [RFC6361] Carlson, J. and Eastlake, D. "PPP Transparent Interconnection of Lots of Links (TRILL) Protocol Control Protocol", RFC 6361, August 2011.
- [RFC6905] Senevirathne, T. et.al. "Requirements for Operations, Administration, and Maintenance (OAM) in Transparent Interconnection of Lots of Links (TRILL)", RFC 6905, March 2013.
- [rfc6326bis] Eastlake, D., Senevirathne, T., Ghanwani, A., Dutt, D., and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", draft-ietf-isis-rfc6326bis, work in progress.
- [RFCclcorrect] Eastlake, Donald, et.al. "TRILL: Clarifications, Corrections, and Updates, draft-ietf-trill-clear-correct, July 2012, in RFC Editor's queue.
- [TRLOAMFRM] Salam, S., et.al., "TRILL OAM Framework", draft-ietf-trill-oam-framework, Work in Progress, November, 2012.
- [TRILLEX] Eastlake, Donald, et.al. "TRILL: Header Extension", draft-ietf-trill-rbridge-extension, June, 2012, in RFC Editor's queue.
- [Y1731] ITU, "OAM functions and mechanisms for Ethernet based networks", ITU-T G.8013/Y.1731, July, 2011.
- [Channel] D. Eastlake, et.al. , "TRILL: RBridge Channel Support", draft-ietf-trill-rbridge-channel-08.txt, in RFC Editor's queue.
- [TRILLOAMMIB] Deepak Kumar et.al, "TRILL OAM MIB", draft-deepak-trill-oam-mib, May 2013, work in progress.

17. Acknowledgments

Work in this document was largely inspired by the directions provided by Stewart Bryant in finding a common OAM solution between SDOs.

Acknowledgments are due for many who volunteered to review this document, notably, Dan Romascanu, Gayle Nobel and Tal Mizrahi.

Special appreciations are due for Dinesh Dutt for his support and encouragement, especially during the initial discussion phase of TRILL OAM.

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

Appendix A.

Backwards Compatibility

Methodology presented above in this document is in-line with the [8021Q] framework for providing fault management coverage. However, in practice, some TRILL platforms may not have the capabilities to support some of the required techniques. In this section, we present a method that allows R Bridges, which do not have the required hardware capabilities, to participate in the TRILL OAM solution.

There are two broad areas to be considered; 1. Maintenance Point (MEP/MIP) Model 2. Data plane encoding and frame identification

A.1 Maintenance Point (MEP/MIP) Model

For backwards compatibility, MEPs and MIPs are located in the CPU. This will be referred to as the "central brain" model as opposed to "port brain" model.

In the "central brain" model, an R Bridge using either ACLs or some other method, forwards qualifying OAM messages to the CPU. The CPU then performs the required processing and multiplexing to the correct MP (Maintenance Point).

Additionally, R Bridges MUST have the capability to prevent the leaking of OAM packets, as specified in [RFC6905].

A.2 Data plane encoding and frame identification

The backwards compatibility method presented in this section defines methods to identify OAM frames when implementations do not have capabilities to utilize TRILL OAM Alert flag presented earlier to identify OAM frames, in the hardware.

It is assumed ECMP path selection of non-IP flows utilize MAC DA, MAC SA and VLAN, IP Flows utilize IP DA, IP SA and TCP/UDP port numbers and other Layer 3 and Layer 4 information. The well-known fields to identify OAM flows are chosen such that they mimic the ECMP selection of the actual data along the path. However, it is important to note that, there may be implementations that would utilize these well-known fields for ECMP selections. Hence, implementations that support OAM SHOULD move to utilizing TRILL Alert Flag, as soon as possible and methods presented here SHOULD be used only as an interim solution.

Identification methods are divided in to 4 broader groups:

1. Identification of Unicast non-IP OAM Flows,
2. Identification of Multicast non-IP OAM Flows,
3. Identification of Unicast IP OAM Flows and
4. Identification of Multicast IP OAM Flows

As presented in the table below, based on the flow type (as defined above), implementations are required to use a well-known value in either the Inner.MacSA field or OAM Ethertype field to identify OAM flows.

Receiving RBridge identifies OAM flows based on the presence of the well-known values in the specified fields, and additionally, for unicast flows, egress RBridge nickname of the packet MUST match that of the local RBridge or for multicast flows, TRILL header multicast flag MUST be set.

Unicast OAM flows that qualify for local processing MUST be redirected to the OAM process and MUST NOT be forwarded (that to prevent leaking of the packet out of the TRILL campus).

A copy of Multicast OAM flows that qualify for local processing MUST be sent to the OAM process and packet MUST be forwarded along the normal path. Additionally, methods MUST be in place to prevent multicast packets leaking out of the TRILL campus.

The following table summarizes the identification of different OAM frames from data frames.

Flow Entropy	Inner MacSA	OAM Ether Type	Egress nickname
unicast no IP	N/A	Match	Match
Multicast no IP	N/A	Match	N/A
Unicast IP	Match	N/A	Match
Multicast IP	Match	N/A	N/A

Figure 22 Identification of TRILL OAM Frames

The unicast and multicast Inner.MacSAs used for the unicast and multicast IP cases, respectively, are TBDmac1 and TBDmac2 assigned by the request in Appendix C.

It is important to note that all RBridges MUST generate OAM flows with "A" flag set and CFM EtherType "0x8902" at the flow entropy off-set. However, well-known values MUST be utilized as part of the flow-entropy when generating OAM messages destined for older RBridges that are compliant to the backwards compatibility method defined in this appendix.

Appendix B.

Base Mode for TRILL OAM

CFM, as defined in [8021Q], requires configuration of several parameters before the protocol can be used. These parameters include MAID, Maintenance Domain Level (MD-LEVEL) and MEPIDs. The Base Mode for TRILL OAM defined here facilitates ease of use and provides out of the box plug-and-play capabilities, supporting the Operational and Manageability considerations described in Section 6 of [TRLOAMFRM].

All RBridges that support TRILL OAM MUST support Base Mode operation.

All Rbridges MUST create a default MA with MAID as specified herein.

MAID [8021Q] has a flexible format and includes two parts: Maintenance Domain Name and Short MA name. In the Based Mode of operation, the value of the Maintenance Domain Name must be the character string "TrillBaseMode" (excluding the quotes "). In Base Mode operation Short MA Name format is set to 2-octet integer format (value 3 in Short MA Format field) and Short MA name set to 65532 (0xFFFC).

The Default MA belongs to MD-LEVEL 3.

[[[Why 3? Doesn't this draft say earlier that TRILL CFM currently just uses level zero?]]]

In the Base Mode of operation, each RBridge creates a single UP MEP associated with a virtual OAM port with no physical layer (NULL PHY). The MEPID associated with this MEP is the 2-octet RBridge Nickname.

By default, all RBridges operating in the Base Mode for TRILL OAM are able to initiate LBM, PT and other OAM tools with no configuration.

Implementations MAY provide default flow-entropy to be included in OAM messages. Content of the default flow-entropy is outside the scope of this document.

Figure 23, below depicts encoding of MAID within CCM messages.

Field Name	Size
Maintenance Domain Format	1
Maintenance Domain Length	2
Maintenance Domain Name	variable
Short MA Name Format	1
Short MA Name Length	2
Short MA Name	variable
Padding	Variable

Figure 23 MAID structure as defined in [8021Q]

Maintenance Domain Name Format is set to Value: 4

Maintenance Domain Name Length is set to value: 13

Maintenance Domain Name is set to: TrillBaseMode

Short MA Name Format is set to value: 3

Short MA Name Length is set to value: 2

Short MA Name is set to : FFFC

Padding : set of zero up to 48 octets of total length of the MAID.

Please refer to [8021Q] for details.

Appendix C.

Unicast MAC Request

Applicant Name: IETF TRILL Working Group

Applicant Email: tsenevir@cisco.com

Applicant Telephone: +1-408-853-2291

Use Name: TRILL OAM

Document: draft-tissa-trill-oam-fm

Specify whether this is an application for EUI-48 or EUI-64

identifiers: EUI-48

Size of Block requested: 1

Specify multicast, unicast, or both: Both

Authors' Addresses

Tissa Senevirathne
CISCO Systems
375 East Tasman Drive.
San Jose, CA 95134
USA.

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

Norman Finn
CISCO Systems
510 McCarthy Blvd
Milpitas, CA 95035
USA

Email: nfinn@cisco.com

Samer Salam
CISCO Systems
595 Burrard St. Suite 2123
Vancouver, BC V7X 1J1, Canada

Email: ssalam@cisco.com

Deepak Kumar
CISCO Systems
510 McCarthy Blvd,
Milpitas, CA 95035, USA

Phone : +1 408-853-9760
Email: dekumar@cisco.com

Donald Eastlake
Huawei Technologies
155 Beaver Street
Milford, MA 01757

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

Sam Aldrin
Huawei Technologies
2330 Central Express Way
Santa Clara, CA 95951
USA

Email: aldrin.ietf@gmail.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56625375
Email: liyizhou@huawei.com

TRILL Working Group
Internet Draft
Intended Status: Standard Track

Expires July 2014

Deepak Kumar
Samer Salam
Tissa Senevirathne
Cisco
January 15, 2014

TRILL OAM MIB
draft-ietf-trill-oam-mib-00.txt

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), 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/ietf/lid-abstracts.txt>.

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

This Internet-Draft will expire on November 08, 2013.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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	3
2. The Internet-Standard Management Framework	3
3. Overview	4
4. Conventions	4
5. Structure of the MIB module	4
5.1. Textual Conventions	4
5.2. TRILL-OAM-MIB relationship to IEEE8021-TC-MIB	4
5.3. TRILL OAM MIB Tree	5
5.3.1. Notifications	5
5.3.2. TRILL OAM MIB Per MEP Objects	5
5.3.2.1. trillOamMepTable Objects	5
5.3.2.2. trillOamMepFlowCfgTable Objects	8
5.3.2.3. trillOamPtrTable Objects	9
5.3.2.4. trillOamMtrTable Objects	10
5.3.2.4. trillOamMepDbTable Objects	12
6. Relationship to other MIB module	13
6.1. Relationship to IEEE8021-CFM-MIB	13
6.2. MIB modules required for IMPORTS	13
7. Definition of the TRILL OAM MIB module	13
8. Security Considerations	47
9. IANA Considerations	48
10. References	48
10.1. Normative References	48
10.2. Informative References	49
11. Acknowledgments	49

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 dotlagCfmMepEntry (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 trillOamMepPtrNextEgressIdentifier - 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 field.

- 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 trillOamMepPtrNextHopNicknames - 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 field of each TLV, concatenated together.

o trillOamMepMtrNextHopNicknames - indicates 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.

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 dotlagCfmMepDbTable, and rows are automatically added or deleted from this table based upon row creation and destruction of the dotlagCfmMepDbTable.

o trillOamMepDbFlowIndex - This object identifies the Flow. If the Flow Identifier TLV is received then 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:

- o dotlagCfmMdIndex
- o dotlagCfmMaIndex
- o dotlagCfmMepIdentifier
- o dotlagCfmMepEntry
- o dotlagCfmMepDbEntry
- o DotlagCfmIngressActionFieldValue
- o DotlagCfmEgressActionFieldValue
- o DotlagCfmRemoteMepState

trillOamMepTable Augments dotlagCfmMepEntry. 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
dotlagCfmMdIndex,
dotlagCfmMaIndex,
dotlagCfmMepIdentifier,
dotlagCfmMepEntry,
dotlagCfmMepDbEntry,
DotlagCfmIngressActionFieldValue,
DotlagCfmEgressActionFieldValue,
DotlagCfmRemoteMepState
    FROM IEEE8021-CFM-MIB
LldpChassisId,
LldpChassisIdSubtype,
LldpPortId
    FROM LLDP-MIB;

trilloamMib MODULE-IDENTITY
    LAST-UPDATED      "201310191200Z"
    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 - Telecommunicatio

n

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            "201310191200Z"
    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 dotlagCfmMepTable and rows
        are automatically added or deleted from this table based upon
        row creation and destruction of the dotlagCfmMepTable."

```

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 { dotlagCfmMepEntry }
 ::= { 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 11.1.1.1"
    ::= { 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 12.2.1"
    ::= { 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 11"
    ::= { 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 11"
 ::= { 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 11"
 ::= { 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 12"
 ::= { 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 12"
 ::= { 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 12"
 ::= { 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 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 10.1.2.1"
 ::= { 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 10.1.2.1"
 ::= { 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 11"
    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 11"
    ::= { 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 11"
    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 11"
    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 11"
    ::= { 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 11"
    ::= { 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 12"
 ::= { 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 12"
 ::= { 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 State
    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 12"
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 12"
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 12"
::= { 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 12"
::= { 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 12"

::= { 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      {
                dotlagCfmMdIndex,
                dotlagCfmMaIndex,
                dotlagCfmMepIdentifier,
                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 {
 dotlagCfmMdIndex,
 dotlagCfmMaIndex,
 dotlagCfmMepIdentifier,
 trilloamMepPtrTransactionId


```

    }
    ::= { trillOamPtrTable 1 }

TrillOamPtrEntry ::= SEQUENCE {
    trillOamMepPtrTransactionId      Unsigned32,
    trillOamMepPtrHC                 Unsigned32,
    trillOamMepPtrFlag               Unsigned32,
    trillOamMepPtrErrorCode          Unsigned32,
    trillOamMepPtrTerminalMep        TruthValue,
    trillOamMepPtrLastEgressId       Unsigned32,
    trillOamMepPtrIngress            DotlagCfmIngressActionFieldValu
e,
    trillOamMepPtrIngressMac          MacAddress,
    trillOamMepPtrIngressPortIdSubtype LldpPortId,
    trillOamMepPtrIngressPortId      LldpPortId,
    trillOamMepPtrEgress             DotlagCfmEgressActionFieldValue
,
    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 11"
    ::= { 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 9.4.3.4"
::= { trillOamPtrEntry 6 }

trillOamMepPtrIngress OBJECT-TYPE
SYNTAX DotlagCfmIngressActionFieldValue
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 9.4.1"
::= { 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 9.4.1"
::= { 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 9.4.1"
::= { 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 9.4.1"
::= { trilloamPtrEntry 10 }

trilloamMepPtrEgress OBJECT-TYPE

SYNTAX DotlagCfmEgressActionFieldValue
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 9.4.1"
::= { 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 9.4.1"
::= { 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 9.4.1"
::= { 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 9.4.1"
::= { 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 9.4.1"
::= { 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 9.4.1"
::= { 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 9.4.1"

::= { 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 9.4.3.5"

::= { 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

```
{
    dotlagCfmMdIndex,
    dotlagCfmMaIndex,
    dotlagCfmMepIdentifier,
    trillOamMepPtrTransactionId,
    trillOamMepMtrReceiveOrder
}
```

```
::= { trillOamMtrTable 1 }
```

```
TrillOamMtrEntry ::= SEQUENCE {
```

```
    trillOamMepMtrTransactionId      Unsigned32,
    trillOamMepMtrReceiveOrder       Unsigned32,
    trillOamMepMtrFlag               Unsigned32,
    trillOamMepMtrErrorCode          Unsigned32,
    trillOamMepMtrLastEgressId       Unsigned32,
    trillOamMepMtrIngress            DotlagCfmIngressActionFieldValu
```

e,

```
    trillOamMepMtrIngressMac         MacAddress,
    trillOamMepMtrIngressPortIdSubtype LldpPortId,
    trillOamMepMtrIngressPortId      LldpPortId,
    trillOamMepMtrEgress             DotlagCfmEgressActionFieldValue
```

,

```
    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 12"
```

```
::= { 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"

::= { 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, 9.4.2.1"

::= { 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, 9.4.2.1"

::= { 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 9.4.3.4"

::= { trilloamMtrEntry 5 }

trilloamMepMtrIngress OBJECT-TYPE

SYNTAX DotlagCfmIngressActionFieldValue

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 12.2.3"

```
::= { 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 12.2.3"
    ::= { 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 12.2.3"
    ::= { 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 12.2.3"
    ::= { trillOamMtrEntry 9 }

trillOamMepMtrEgress OBJECT-TYPE
    SYNTAX          DotlagCfmEgressActionFieldValue
    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 12.2.3"
    ::= { 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 12.2.3"
::= { 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 12.2.3"
    ::= { 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 12.2.3"
    ::= { 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 12.2.3"
    ::= { 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 12.2.3"
    ::= { 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 12.2.3"
::= { 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 9.4.3.5"
::= { 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 9.4.3.6"
::= { 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 9.4.3.6"
::= { 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 dotlagCfmMepDbTable and rows are automatically added or deleted from this table based upon row creation and destruction of the dotlagCfmMepDbTable."

REFERENCE

"[TRILL-FM]"

::= { trillOamMep 5 }

trillOamMepDbEntry OBJECT-TYPE

SYNTAX TrillOamMepDbEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The conceptual row of trillOamMepDbTable."

AUGMENTS {
dotlagCfmMepDbEntry
}

::= { trillOamMepDbTable 1 }

TrillOamMepDbEntry ::= SEQUENCE {

trillOamMepDbFlowIndex	Unsigned32,
trillOamMepDbFlowEntropy	OCTET STRING,
trillOamMepDbFlowState	DotlagCfmRemoteMepState,
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      DotlagCfmRemoteMepState
    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:

dotlagCfmMdIndex - Also the index of the MEP's
 Maintenance Domain table entry
 (dotlagCfmMdTable).

dotlagCfmMaIndex - Also an index (with the MD table index)
 of the MEP's Maintenance Association
 network table entry
 (dotlagCfmMaNetTable), and (with the MD
 table index and component ID) of the
 MEP's MA component table entry
 (dotlagCfmMaCompTable).

dotlagCfmMepIdentifier - MEP Identifier and final index
 into the MEP table (dotlagCfmMepTable).

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 MAX-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 SMIPv2", 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, February, 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.

Copyright (c) 2014 IETF Trust and the persons identified as authors of the code. All rights reserved. Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

Copyright (c) 2014 IETF Trust and the persons identified as authors of the code. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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: August 2, 2014

M. Wasserman
Painless Security
D. Eastlake
D. Zhang
Huawei Technologies
January 31, 2014

Transparent Interconnection of Lots of Links (TRILL) over IP
draft-mrw-trill-over-ip-04.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 August 2, 2014.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements Terminology	2
2. Introduction	3
3. Use Cases for TRILL over IP	3
3.1. Remote Office Scenario	3
3.2. IP Backbone Scenario	4
3.3. Important Properties of the Scenarios	4
3.3.1. Security Requirements	4
3.3.2. Multicast Handling	5
3.3.3. RBridge Neighbor Discovery	5
4. TRILL Packet Formats	5
4.1. TRILL Data Packet	5
4.2. TRILL IS-IS Packet	6
5. Link Protocol Specifics	6
6. Port Configuration	7
7. TRILL over UDP/IP Format	7
8. Handling Multicast	8
9. Use of DTLS	8
10. Transport Considerations	9
10.1. Recursive Ingress	9
10.2. Fat Flows	10
10.3. Congestion Considerations	10
11. MTU Considerations	10
12. Middlebox Considerations	11
13. Security Considerations	11
14. IANA Considerations	12
15. Acknowledgements	12
16. References	13
16.1. Normative References	13
16.2. Informative References	14
Authors' Addresses	14

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] [I-D.eastlake-isis-rfc6326bis] [I-D.ietf-trill-rfc6327bis].

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 [I-D.ietf-trill-fine-labeling] 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] 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[I-D.ietf-trill-fine-labeling] 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. The 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 [I-D.ietf-trill-rfc6327bis], and to establish authenticated communication with those RBridges.

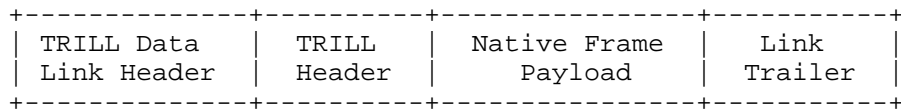
In the Remote Office Scenario, a DTLS session will need to be established between RBridges before TRILL IS-IS traffic can be exchanged, as discussed below. In this case, one of the RBridges will need to be configured to establish a DTLS session with the other RBridge. This will typically be accomplished 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

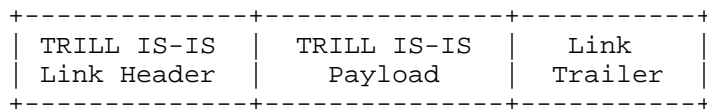
The on-the-wire form of a TRILL Data packet in transit between two neighboring RBridges is as shown below:



Where the Encapsulated Native Frame is similar to Ethernet frame format with a VLAN tag or Fine Grained Label [I-D.ietf-trill-fine-labeling] 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

[I-D.eastlake-isis-rfc6326bis][I-D.ietf-trill-rfc6327bis] on the link. This synthetic SNPA is as follows:

```

  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-----+-----+-----+-----+
| 0xFE          | 0x00          |
+-----+-----+-----+-----+
| IPv4 upper half          |
+-----+-----+-----+-----+
| IPv4 lower half         |
+-----+-----+-----+-----+

```

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.

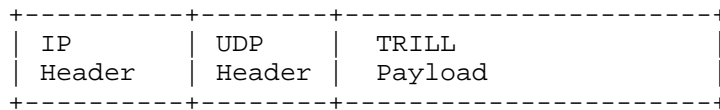
6. Port Configuration

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.

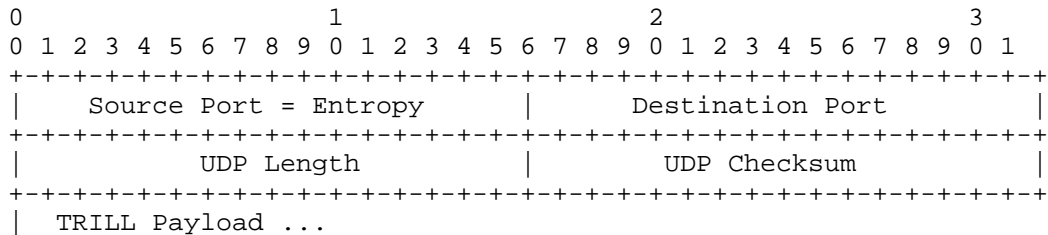
TBD: MUST be able to configure a list of IP addresses for serial unicast. MUST be able to configure a non-standard IP multi-cast address if native multicast is being used.

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 unicast addresses of other stations to which multi-destination packets are sent.

TBD

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, the DTLS session MUST be fully established before any TRILL IS-IS or data frames are exchanged.

RBridges that implement TRILL over IP SHOULD support the use of certificates for DTLS and, if they support certificates, MUST support the following algorithm:

- o TLS_RSA_WITH_AES_128_CBC_SHA [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:

- o TLS_PSK_WITH_AES_128_CBC_SHA [RFC5246]

If the derived default preshared key is used, it 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.

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 be able to test whether a TRILL packet they are about 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 could be 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] [I-D.eastlake-isis-rfc6326bis]. 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 ports is described in [I-D.ietf-trill-clear-correct]. 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 [I-D.ietf-trill-rfc6327bis] 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 where 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.
- [I-D.eastlake-isis-rfc6326bis]
Eastlake, D., Senevirathne, T., Ghanwani, A., Dutt, D., and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", draft-eastlake-isis-rfc6326bis-09 (work in progress), August 2012.
- [I-D.ietf-trill-clear-correct]
Eastlake, D., Zhang, M., Ghanwani, A., Manral, V., and A. Banerjee, "TRILL: Clarifications, Corrections, and Updates", draft-ietf-trill-clear-correct-06 (work in progress), July 2012.
- [I-D.ietf-trill-rfc6327bis]
Eastlake, D., Perlman, R., Ghanwani, A., Yang, H., and V. Manral, "TRILL: Adjacency", draft-ietf-trill-rfc6327bis-03 (work in progress), January 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [RFC6325] Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

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 K. Building, "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.

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: Informational

Yizhou Li
Donald Eastlake
Weiguo Hao
Huawei Technologies
Radia Perlman
Intel Labs
Jon Hudson
Brocade
Hongjun Zhai
ZTE
February 14, 2014

Expires: August 18, 2014

Problem Statement and Goals for Active-Active TRILL Edge
draft-yizhou-trill-active-active-connection-prob-02

Abstract

The IETF TRILL (Transparent Interconnection of Lots of Links) protocol provides support for flow level multi-pathing with rapid failover for both unicast and multi-destination traffic in networks with arbitrary topology between TRILL switches. Active-active at the TRILL edge is the extension of these characteristics to end stations that are multiply connected to a TRILL campus. This informational document discusses the high level problems and goals when providing active-active connection at the TRILL edge.

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/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at

<http://www.ietf.org/shadow.html>

Copyright and License Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	3
1.1	Terminology	3
2.	Target Scenario	3
3.	Problems in Active-Active at the TRILL Edge	6
3.1	Frame Duplications	6
3.2	Loop	6
3.2	Address Flip-Flop	6
3.3	Unsynchronized Information Among Member RBridges	7
4	High Level Requirements and Goals for Solutions	7
5	Security Considerations	8
6	IANA Considerations	8
7	References	8
7.1	Normative References	8
7.2	Informative References	9
	Authors' Addresses	9

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. In a TRILL campus, communications between TRILL switches can

(1) use multiple parallel links and/or paths,

(2) load spread over different links and/or paths at a fine grained flow level through equal cost multipathing of unicast traffic and multiple distribution trees for multi-destination traffic, and

(3) rapidly re-configure to accommodate link or node failures or additions.

"Active-active" is the extension, to the extent practical, of similar load spreading and robustness to the connections between end stations and the TRILL campus. Such end stations may have multiple ports and will be connected, directly or via bridges, to multiple edge TRILL switches. It must be possible, except in some failure conditions, to load spread end station traffic at the flow level across links to such multiple edge TRILL switches and rapidly re-configure to accommodate topology changes.

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

CE - customer equipment. Could be a bridge or end station or a hypervisor.

Edge group - a group of edge RBridges to which at least one CE is multiply attached. One RBridge can be in more than one edge group.

TRILL switch - an alternative term for an RBridge.

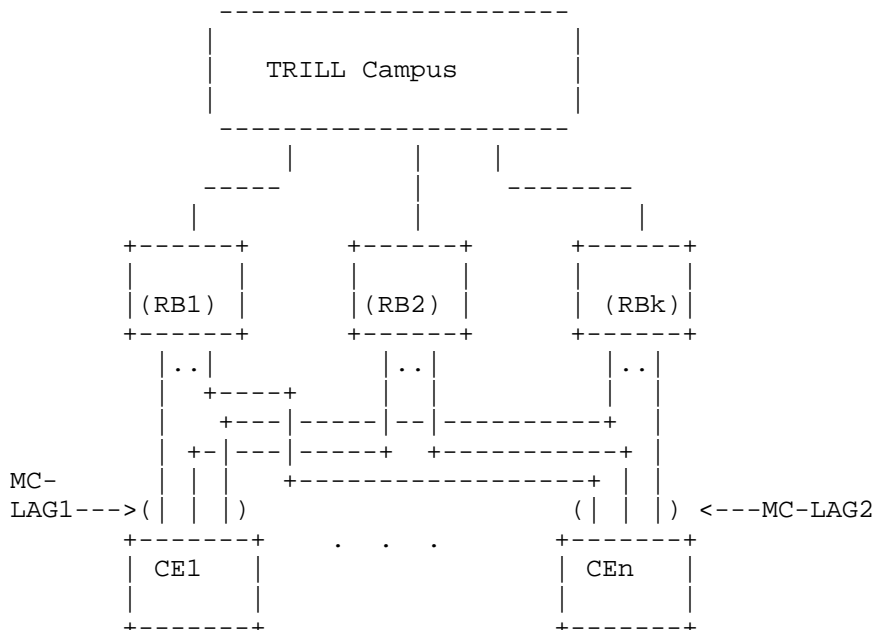
2. Target Scenario

The TRILL appointed forwarder [RFC6325] [RFC6327bis] [RFC6439]

mechanism provides per VLAN active-standby traffic spreading and loop avoidance at the same time. One and only one appointed RBridge can ingress/egress native frames into/from TRILL campus for a given VLAN among all edge RBridges connecting a legacy network to TRILL campus. This is true whether the legacy network is a simple point-to-point link or a complex bridged LAN or anything in between. By carefully selecting different RBridge as appointed forwarder for different set of VLANs, load spreading over different edge RBridges across different VLANs can be achieved.

This section presents a typical scenario of active-active connections to TRILL campus via multiple edge RBridges where the current TRILL appointed forwarder mechanism is not applicable.

The appointed forwarder mechanism [RFC6439] requires each of the edge RBridges to exchange TRILL IS-IS Hello packets from their access ports. As figure 1 shows, when multiple access links of multiple edge RBridges are bundled as an MC-LAG (Multi-Chassis Link Aggregation Group), Hello messages sent by RB1 via access port to CE1 will not be forwarded to RB2 by CE1. RB2 (and other members of MC-LAG1) will not see that Hello from RB1. Every member RBridge of MC-LAG1 thinks of itself as appointed forwarder on MC-LAG1 link for all VLANs and will ingress/egress frames for all VLANs. Hence the appointed forwarder mechanism is not applicable in such an active-active scenario.



Active-Active connection is useful when we want to achieve the following goals.

- Flow rather than VLAN based load balancing is desired.
- More rapid failure recovery is desired. Current appointed forwarder mechanism relies on the Hello timer expiration to detect the unreachability of another edge RBridge connecting to the same local Ethernet link. Then re-appointing the forwarder for specific VLANs may be required. Such procedures takes time in the scale of seconds. Active-Active connection usually has faster built-in mechanism for member node and/or link failure detection. Faster detection of failure would minimize the frame loss and recovery time.

MC-LAG is a proprietary facility whose implementation varies by vendor. So, to be sure of MC-LAG operation across an edge group of RBridges, those edge RBridges will almost always be from the same vendor. In order to have common understanding of active-active connection scenarios, the following assumptions are made:

For CE connecting to multiple edge RBs via active-active connection:

- a) the CE will forward a packet from an endnode to exactly one up-link
- b) the CE will never forward packets it receives from one up-link to another
- c) the CE will attempt to send all packets for a given flow on the same uplink
- d) packets are accepted from any of the uplinks and passed down to endnodes (if any exist)
- e) the CE has some unknown rule for which packets get sent to which uplinks (typically based on a simple hash function of Layer 2 through 4 header fields)
- f) the CE cannot be assumed to give useful control information to the up-link such as "this set of other RBridges CE is attached", or "these are all the MAC addresses attached"

For an edge group to which a CE is multiply attached:

- a) Any two RBs in the edge group are reachable from each other
- b) Each RB in the edge group is configured with a name for each down-link to an CE multiply attached to that group. The names will be consistent across the edge group. For instance, if CE1 attaches to RB1, RB2 to RBn, then each of RBs will have been configured, for the port to CE1, that it is labeled "MC-LAG1"
- c) The RBs in the edge group have existing mechanisms to exchange states and information with each other, including the set of CEs they are connecting to or name of MC-LAGs their down-links have joined
- d) Each RB in the edge group can be configured with the set of

acceptable VLANs (or fine-grained labels) for the ports to any CE. The acceptable VLANs configured for those port should include all the VLANs the CE has joined and be consistent for all the member RB.

e) When a RB fails, all the other RBs having formed any MC-LAG with it know the information timely

f) When a down-link of a RB fails, all the other RBs having formed any MC-LAG with that down-link know the information timely

3. Problems in Active-Active at the TRILL Edge

This section presents the problems that need to be addressed in active-active connection scenarios. The topology in Figure 1 is used in the following sub-sections as the example scenario for illustration purposes.

3.1 Frame Duplications

When a remote RBridge sends a multi-destination TRILL Data packet in VLAN x, all member RBridges of MC-LAG1 will receive the frame if any local CE1 joins VLAN x. As each of them thinks it is the appointed forwarder for all VLANs, without active-active changes they would all forward the frame to CE1. The bad consequence is that CE1 receives multiple copies of that multi-destination frame from the remote end host.

It should be noted frame duplication is only a problem in multi-destination frame forwarding. Unicast forwarding does not have this issue.

3.2 Loop

As shown in Figure 1, CE1 may send a native multi-destination frame to TRILL campus via a member of MC-LAG1 (say RB1). This frame will be TRILL encapsulated and then forwarded through the campus to another member (say RB2) of the same MC-LAG. In this case, without active-active changes RB2 will decapsulate the frame and forward it. The frame loops back to CE1.

3.2 Address Flip-Flop

Consider RB1 and RB2 using their own nickname as ingress nickname for data into a TRILL campus. As shown by Figure 1, CE1 may send a data frame with the same source VLAN/MAC address to any member of the edge group MC-LAG1. If the egress RBridge receives TRILL data packets from different ingress RBridges but with same source VLAN/MAC address, it learns different address correspondence from the decapsulated data frames. Address correspondence may keep flip-flopping among nicknames of the member RBridges of the MC-LAG for the same MAC address in the

same VLAN.

Most TRILL switches may behave badly under these circumstances and, for example, interpret this as a severe network problem. It may also cause the returning traffic to go through the different paths to reach the destination resulting in persistent re-ordering of the frames.

3.3 Unsynchronized Information Among Member RBridges

A local Rbridge, say RB1 in MC-LAG1, may have learned a VLAN/MAC and 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 MC-LAG1, e.g., RB2. RB2 may not have h1's VLAN/MAC and nickname 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.

Synchronization on the VLAN/MAC and nickname correspondence information among member RBridges will reduce such unnecessary flooding.

Unsynchronized multicast group information causes problems too. The edge RBridge snoops the IGMP [RFC3376] join message from CE may not be the one receiving the multicast traffic for the joined group later. Therefore multicast traffic can be dropped incorrectly.

TRILL [RFC6325] designed its multi-destination traffic forwarding with some specific mechanisms, e.g., Reverse Path Forwarding Check, tree calculation, construction and selection, pruning, etc. Solutions of active-active connection at edge RBridges should carefully examine those features and make sure they work correctly.

4 High Level Requirements and Goals for Solutions

Problems identified in section 3 should be solved in any solution for active-active connection to RBridges. The requirements are summarized as follows,

- a) Loop and frame duplication MUST be prevented
- b) Learning of VLAN/MAC and nickname correspondence by a remote RBridge MUST not flip-flop between the local multiply attached edge RBridges
- c) Member RBridges of an MC-LAG MUST be able to share relevant TRILL specific information with each other

In addition, the following high level goals should be met also.

Data plane:

- 1) all up-links of CE MUST be active. CE is free to choose any up-link on which to send packets
- 2) packets for a flow should stay in order
- 3) the Reverse Path Forwarding Check MUST work properly as per [RFC6325]
- 4) Single up-link failure on CE to an edge group MUST not cause persistent packet delivery failure between TRILL campus and CE

Control plane:

- 1) no requirement for new information to be passed between edge RBridges and CE
- 2) If there are any TRILL specific parameters required to be exchanged between RBridges in an edge group, e.g., nicknames, solution SHOULD specify the mechanism to perform such exchange.

Configuration, incremental deployment and others:

- 1) Solution should require minimal configuration
- 2) Solution should automatically detect misconfiguration of edge RBridge group
- 3) Solution should support incremental deployment, i.e. not require campus wide upgrading for all RBridges, only changes to the edge group RBridges
- 4) Solution should be able to support at least 4 active-active up-links on a multiply attached CE

5 Security Considerations

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

6 IANA Considerations

No IANA action is required. RFC Editor: please delete this section before publication.

7 References

7.1 Normative References

- [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.

- [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
- [RFC6326bis] Eastlake, D., Banerjee, A., Dutt, D., Perlman, R., and A. Ghanwani, "TRILL Use of IS-IS", draft-eastlake-isis-rfc6326bis, work in progress.
- [RFC6327bis] Eastlake 3rd, D., R. Perlman, A. Ghanwani, H. Yang, and V. Manral, "TRILL: Adjacency", draft-ietf-trill-rfc6327bis, work in progress.
- [RFC6439] Perlman, R., Eastlake, D., Li, Y., Banerjee, A., and F. Hu, "Routing Bridges (RBridges): Appointed Forwarders", RFC 6439, November 2011

7.2 Informative References

- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [TRILLPN] Zhai, H., et.al., "RBridge: Pseudonode Nickname", draft-hu-trill-pseudonode-nickname, Work in progress, November 2011.
- [8021AX] IEEE, "Link Aggregation", 802.1AX-2008, 2008.
- [8021Q] IEEE, "Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q-2011, August, 2011

Authors' Addresses

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56625409
EMail: liyizhou@huawei.com

Donald Eastlake

Huawei Technologies
155 Beaver Street
Milford, MA 01757 USA

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

Weiguo Hao
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56623144
EMail: haoweiguo@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

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

Phone: +1-408-333-4062
jon.hudson@gmail.com

Hongjun Zhai
ZTE
68 Zijinghua Road, Yuhuatai District
Nanjing, Jiangsu 210012
China

Phone: +86 25 52877345
Email: zhai.hongjun@zte.com.cn

INTERNET-DRAFT
Intended Status: Proposed Standard

Mingui Zhang
Huawei
Radia Perlman
Individual Contributor
Hongjun Zhai
ZTE
Mukhtiar Shaikh
Brocade
February 14, 2014

Expires: August 18, 2014

TRILL Active-Active Edge Using Multiple MAC Attachments
draft-zhang-trill-aa-multi-attach-00.txt

Abstract

TRILL active-active service is to provide end stations with flow level load balance and resilience against link failures at the edge of TRILL campuses.

This draft proposes that member RBridges in an active-active edge RBridge group use their own nickname as the ingress RBridge nickname to encapsulate frames from attached end systems. Thus, remote edge RBridges are required to learn multiple locations of one MAC address in one VLAN. Design goals of this proposal 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/lid-abstracts.html>

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

Copyright and License Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Acronyms and Terminology	3
2.1. Acronyms	3
2.2. Terminology	4
3. Overview	4
4. Backward Compatibility	5
5. Design Goals	5
5.1. No MAC Flip-Floping (Normal Unicast Egress)	6
5.2. Regular Unicast/Multicast Ingress	6
5.3. Right Multicast Egress	6
5.3.1. No Duplication (Single Exit Point)	6
5.3.1. No Echo (Split Horizon)	6
5.4. No Black-hole & No Triangular Forwarding	7
5.5. Load Balance Towards the AAE	7
6. Contributors	7
7. Security Considerations	7
8. IANA Considerations	7
Acknowledgements	8
9. References	8
9.1. Normative References	8
9.2. Informative References	8
Author's Addresses	10

1. Introduction

In the TRILL Active-Active Edge (AAE) topology, a Multi-Chassis Link Aggregation Group (MC-LAG) is used to connect multiple R Bridges to a switch or a vSwitch. An endnode clump is attached to this switch or vSwitch. It's required that data traffic within a specific VLAN from this endnode clump can be ingressed and egressed by any of these R Bridges simultaneously. End systems in the clump can spread their traffic among these edge R Bridges at the flow level. When a link fails, end systems can keep using the rest of links in the MC-LAG, which provides the resilience towards link failures.

Since a packet from each endnode can be ingressed by any R Bridge in the AAE group, a remote edge R Bridge may observe multiple attachment points (i.e., egress R Bridges) for this endnode identified by its MAC address. This issue is known as the "MAC flip-flopping" issue. Three potential solutions arise to address this issue:

- 1) AAE member R Bridges use a pseudonode nickname, instead of their own, as the ingress nickname for end systems attached to the MC-LAG. [CMT] is based on this solution.

- 2) AAE member R Bridges split work among themselves for which ones will be responsible for which MAC addresses. A member R Bridge will encapsulate the packet using its own nickname if it is responsible for the source MAC address. Otherwise, if the frame is known unicast, it encapsulates the packet using the nickname of the responsible R Bridge; if the frame is multicast, it needs to redirect the packet to its responsible R Bridge for encapsulation.

- 3) AAE member R Bridges keep using their own nicknames. Remote edge R Bridges are required to learn multiple points of attachment per VLAN for a MAC address attached to the AAE, and separately time each one out.

The purpose of this ID is to develop an approach based on solution 3. Although it focuses on exploring solution 3, the major design goals discussed here are common for AAE. Through mirroring the scenarios studied in this draft, other potential solutions may benefit as well.

The main body of the document is organized as follows. Section 2 lists the acronyms and terminologies. Section 3 gives the overview model. Section 4 gives three options for incremental deployment. Section 5 how this approach meets the design goals.

2. Acronyms and Terminology

2.1. Acronyms

TRILL: TRansparent Interconnection of Lots of Links

AAE: Active/Active Edge

MC-LAG: Multi-Chassis Link Aggregation Group

IS-IS: Intermediate System to Intermediate System

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], [RFC6327], [6327bis] and [RFC6439] is assumed in this document.

3. Overview

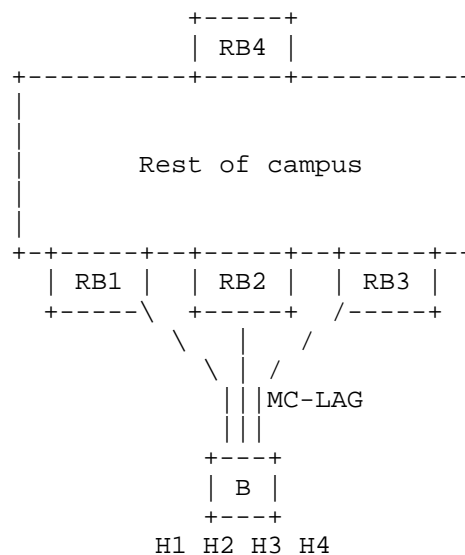


Figure 3.1: An example topology of TRILL Active-Active Edge

Figure 3.1 shows an example network of TRILL Active-Active Edge. In this figure, endnodes (H1, H2, H3 and H4) are attached to a bridge (B) which communicates with multiple RBridges (RB1, RB2 and RB3) via the MC-LAG. Suppose RB4 is a 'remote' RBridge out of the AAE group in the TRILL campus. This connection model is also applicable to the virtualized environment where the hard bridge can be replaced with a vSwitch while those bare metal hosts are replaced with virtual machines (VM).

For packets received from their attached endnode clumps, member

RBridges of the AAE group always encapsulate using their own nickname no matter it's unicast or multicast.

In this proposal, all edge RBridges in the entire campus need to learn multiple attachment points for each MAC address, and separately time each one out.

4. Backward Compatibility

Three options are listed below to cope with incremental deployment scenarios.

-- Option A

A new capability announcement would appear in LSPs. "I can cope with multiple endnode attachments". Only if all edge RBridges announce this capability can the AAE group use this approach. For those legacy RBridges who are not capable to cope with multiple endnode attachments, new type TRILL switches will not establish connectivity with them so that they are isolated from these new type TRILL switches. Note only edge RBridges (those that are Appointed Forwarders [RFC6439]) need to be able to support this. It does not affect totally transit RBridges.

-- Option B

Each edge RBridge in the AAE group ingress data frames from any MC-LAG into a specific topology. In this way, the topology ID is used as the discriminator of different locations of a specific MAC address at the remote RBridge. TRILL MAY reserve a list of topology IDs to be dedicated to AAE. RBridges which do not support this reserved list MUST NOT establish connectivity with edge RBridges in the AAE group.

-- Option C

If the data plane learning of all RBridges does not support the multiple locations learning feature. It's possible to make use of the ESADI protocol [ESADI] to distribute MAC addresses. Compared to the data plane learning, TRILL ESADI allows one RBridge to remember multiple locations of a MAC address at the control plane.

5. Design Goals

Proposals for the major design goals of AAE are explored in this section.

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

Since all RBridges talking with the AAE RBridges in the campus are able to keep multiple locations for one MAC address, a MAC address learnt from one AAE member will not be overwritten by the same MAC address learnt from another AAE member. Multiple entries for this MAC address will be created. The remote RBridge can adhere to one of the locations (e.g., the closest one) for each MAC address rather than keep flip-floping among them.

5.2. Regular Unicast/Multicast Ingress

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

5.3. Right Multicast Egress

The design goal is that there is no duplication and forwarding loop.

5.3.1. No Duplication (Single Exit Point)

When multi-destination packets for a specific VLAN are received from the campus, it's important that exactly one RBridge out of the AAE group let through each multicast packet, therefore no duplication happens. The single exit point can be selected based on static algorithms, e.g., VLAN or source MAC address 'mod' the number of AAE members.

5.3.1. No Echo (Split Horizon)

When a multicast frame originated from an MC-LAG is ingressed by an RBridge of an AAE group, forwarded across 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 MC-LAG. Otherwise, it will cause a forwarding loop (echo). The well known 'split horizon' technique can be used to eliminate the echo issue. The essential point for split horizon is that the MC-LAG is appointed with an unique identifier across the AAE group. When an AAE member receives a multicast packet has this identifier, the receiver MUST NOT egress it to the MC-LAG with the same identifier.

This document propose to split horizon based on the tuple consisting of the Fine Grained Label (FGL) plus the ingress RBridge nickname. When there are multiple MC-LAGs connected to the same RBridge, each

MC-LAG MUST be assigned with an unique FGL. RBridges in an AAE group should discover and remember nicknames of other members. If a multicast packet is from an edge RBridge in a same AAE group as RB1, its FGL will be read and RB1 MUST NOT egress it out of the interface configured with the same FGL. Otherwise, RB1 SHOULD egress the packet without the split horizon behavior.

5.4. No Black-hole & No Triangular Forwarding

If a sub-link of the MC-LAG fails while remote RBridges continue to send packets to those MAC addresses they have learnt via the failed port, black-hole happens.

The proposal in this draft may make use of MAC withdrawal. When a member RBridge detects that the port connected to a sub-link of the MC-LAG fails, all MAC addresses attached to this RBridge through the failed sub-link will be flushed. After doing that, no traffic will be sent via the failed port, hence no black-hole happens.

5.5. Load Balance Towards the AAE

Since a remote RBridge can record multiple attachments of one MAC address, this remote RBridge can choose to spread the traffic to this MAC towards any of the AAE members. Each of them is able to egress the traffic. Flow-level load balance mechanisms can be implemented to optimize the distribution of the traffic load towards the AAE group.

6. Contributors

Muhammad Durrani
Brocade
Email: mdurrani@brocade.com

7. Security Considerations

Security issue should be considered when a specific extension is made to the existing TRILL control plane.

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

For security considerations pertain to extensions hosted by TRILL ESADI and Channel should refer to the Security Considerations in [ESADI] and [Channel].

8. IANA Considerations

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

Acknowledgements

The authors would like to thank the comments and suggestions from Donald Eastlake, Erik Nordmark, Fangwei Hu and Liang Xia.

9. References

9.1. Normative References

- [RFC6325] Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.
- [RFC6327] Eastlake 3rd, D., Perlman, R., Ghanwani, A., Dutt, D., and V. Manral, "Routing Bridges (RBridges): Adjacency", RFC 6327, July 2011.
- [6327bis] D. Eastlake, R. Perlman, et al, "TRILL: Adjacency", draft-ietf-trill-rfc6327bis-04.txt, January 2014, in RFC Ed Queue.
- [RFC6439] Perlman, R., Eastlake, D., Li, Y., Banerjee, A., and F. Hu, "Routing Bridges (RBridges): Appointed Forwarders", RFC 6439, November 2011.
- [ESADI] H. Zhai, F. Hu, et al, "TRILL (Transparent Interconnection of Lots of Links): ESADI (End Station Address Distribution Information) Protocol", draft-ietf-trill-esadi-05.txt, February 2014, working in progress.
- [6326bis] D. Eastlake, T. Senevirathne, et al, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", draft-ietf-isis-rfc6326bis-03.txt, January 2014, in RFC Ed Queue.
- [Channel] D. Eastlake, V Manral, et al, "TRILL: RBridge Channel Support", draft-ietf-trill-rbridge-channel-08.txt, July 2012, working in progress.
- [tunnel] D. Eastlake, Y. Li, "TRILL: RBridge Channel Tunnel Protocol", draft-ietf-trill-channel-tunnel-00.tx, December 2013, working in progress.

9.2. Informative References

- [CMT] T. Senevirathne, J. Pathangi, et al, "Coordinated Multicast Trees (CMT)for TRILL", draft-ietf-trill-cmt-02.txt, November 2012, working 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.

Author's Addresses

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

Email: zhangmingui@huawei.com

Radia Perlman
Individual Contributor

Email: radiaperlman@gmail.com

Hongjun Zhai
ZTE Corporation
68 Zijinghua Road
Nanjing 200012 China

Phone: +86-25-52877345
Email: zhai.hongjun@zte.com.cn

Mukhtiar Shaikh
Brocade

Email: mshaikh@brocade.com