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

Copyright Notice

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

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.

```
       \              +-------+         +------+ TRILL Domain/
        \           +/------+ |       +/-----+ |            /
         \          | Aggr11| + ----- |AggrN1| +           /
          \         +---+---+/        +------+/           /
           \         /     \            /      \         /
            \       /       \          /        \       /
             \   +---+    +---+      +---+     +---+   /
              \- |T11|... |T1x|      |T21| ..  |T2y|---
                 +---+    +---+      +---+     +---+
                   |        |          |         |
                 +-|-+    +-|-+      +-|-+     +-|-+
                 |   |... | V |      | V | ..  | V |<-VirtualSwitch
                 +---+    +---+      +---+     +---+
                 |   |... | V |      | V | ..  | V |
                 +---+    +---+      +---+     +---+
                 |   |... | V |      | V | ..  | V |
                 +---+    +---+      +---+     +---+
```

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.

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.

```
       +------------+--------+---------+---------+--+-------+----+
       |OuterEtherHd|TRILL HD| InnerDA | InnerSA |..|Payload| FCS|
       +------------+--------+---------+---------+--+-------+----+
               ^
               |             |<Inner Ether Header>  |
               |
               |
               |      +-------+  TRILL    +------+
               |      |  R1   |-----------|  R2  |  Decapsulate
   TRILL
               |      +---+---+  domain   +------+  header
               |          |                   |
               +----------|                   |
                          |                   |
                     +-----+             +-----+
      Non-RBridge node:|T12  |             | T22 |
      Encapsulate TRILL+-----+             +-----+
      Header for data
      Frames to traverse
      TRILL domain.
```

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 RBridges receiving TRILL encapsulated frames with their own nickname in the frames’ source address field from neighboring non-RBridge nodes, a new nickname can be given to an RBridge edge node, e.g. Phantom Nickname, to represent all the TRILL Encapsulating Nodes attached to the RBridge 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*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.

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

[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

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

Disclaimer of Liability

Acknowledgment

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

INTERNET-DRAFT
Intended status: Proposed Standard

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

Expires: January 14, 2014
July 15, 2013

TRILL: Directory Assistance Mechanisms
<draft-dunbar-trill-scheme-for-directory-assist-05.txt>

Abstract

This document describes mechanisms for using directory server(s) to assist TRILL (Transparent Interconnection of Lots of Links) edge switches in reducing multi-destination traffic, particularly ARP/ND and unknown unicast flooding.

Status of This Memo

Table of Contents

1. Introduction

[DirectoryFramework] describes a high level framework 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 aginst address spoofing within a TRILL comapus. Because multi-destination traffic becomes an increasing burden as a network scales, 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 mechanims are optional to implement.

The information held by the directories is address mapping information. Most commonly, what MAC address [RFC5342bis] corresponds to an IP address within a Data Label (VLAN or FGL (Fine Grained Label [RFCfgl])) and what egress TRILL switch (RBridge) that MAC address is attached to. But it could be what IP address corresponds to a MAC address or possibly other mappings. In the data center environment, it is common for orchestration software to know and control where all the IP addresses, MAC address, and VLANs/tenants are. Thus such orchestration software is appropriate for providing the directory function or for supplying the Directory(s) with information they need.

Directory services can be offered in a Push or Pull mode. Push mode, in which a directory server pushes information to RBridges indicating interest, is specified in Section 2. Pull mode, in which an RBridge queries a server for the information it wants, is specified in Section 3. Modes of operation including hybrid Push/Pull are discussed in Section 4.

The mechanisms used to keep the mappings held by different Directories synchronized is beyond the scope of this document.

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 additional acronyms and terms:

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.

RBridge: An alternative name for a TRILL switch.

TRILL switch: An alternative name for an RBridge.

1.2 Circumstances Causing Directory Use

An RBridge can consult Directory information whenever it wants, by searching through information it has because that information has been pushed to it or pulled by it and retained or by requesting information from a pull directory. However, the following are expected to be the most common circumstances leading to directory use. All of these are cases of ingressing a native frame.

- o Ingressing a native frame with an unknown unicast destination MAC. The mapping from the destination MAC and Data Label to its egress RBridge of attachment is needed to ingress the frame as unicast. If the egress RBridge is unknown, the frame must be dropped or ingressed as a multi-destination frame and flooded to all edge RBridges for its Data Label.

- o Ingressing an ARP [RFC826]. ARP is a very flexible protocol but is primarily used on a link to query for the MAC address corresponding to an IPv4 address, test if an IPv4 address is in use, or to announce a change in any of IPv4 address, MAC address, and point of attachment. ... more TBD ...

- o Ingressing a ND [RFC903]. ...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.

- o Ingressing a RARP [RFC4861]. ...TBD...

Any of the above could be cause for an ingress RBridge to consult Directory information that has been pushed to it, to send a pull request to a Pull Directory, or both.

2. Push Model Directory Assistance Mechanisms

In the Push Model, Push Directory servers push down the mapping information for the various addresses of end stations in some Data Label. A Push Directory advertises whether or not it believes it is pushing complete mapping information for a Data Label. The Push Model uses the [ESADI] protocol.

With this model, it is RECOMMENDED that complete address mapping information for a Data Label be pushed and that a participating RBridge simply drop a data packet, instead of flooding the packet, if the destination unicast MAC address is in a Data Label being pushed and can’t be found in the address mapping information available. This will minimize flooding of packets due to errors or inconsistencies but is not practical if directories have incomplete information.

2.1 Requesting Push Service

In the Push Model, it is necessary to have a way for an RBridge to request information from the directory server(s). RBridges simply use the ESADI protocol mechanism to announce, in the IS-IS link state database, all the Data Labels for which they are participating in [ESADI]. They are then pushed the mapping information for all such Data Labels being served by a Push Directory server.

2.2 Push Directory Servers

Push Directory servers advertise their availability to push the mapping information for a particular Data Label to ESADI participants for that Data Label by turning on a flag bit in their ESADI Parameter APPsub-TLV [ESADI] (see Section 7.1) for athat ESADI instance.

Each Push Directory server MUST participate in ESADI for the Data Labels for which it can push mappings and set the PD bit in their ESADI-Parameters APPsub-TLV for that Data Label.

2.3 Multiple Push Directory Servers

For robustness, it is useful to have more than one copy of the data being pushed. Each RBridge that is a 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. This is the number of copies of the directory it believes should be pushed.

Each Push Directory server also has an 8-bit priority to be active (see Section 6.1 of this document). This priority is treated as an unsigned integer where larger mangitude means higher priority and is in its ESADI Parameter APPsub-TLV. In cases of equal priority, the 6-byte IS-IS System ID is 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 RBridge server orders the Push Directory servers that it can see as data reachable [RFCclear] in the ESADI link state database for that Data Label and determines its position in that order. If a Push Directory server believes that N copies of the mappings for a Data Label should be pushed and finds that it is first in priority or, more generally, equal to or higher than Nth in priority, it is Active. If it finds that it is N+1st or lower in priority, it is Passive.

For example, assume four Push Directory servers for Data Label X: server A with priority 123 configured to believe there should be 2 copies pushed; server B, priority 88, 1 copy; server C, priority 40, 3 copies; and server D, priority 7, 2 copies. Server A, seeing that is highest priority, is Active. Server B, seeing that it is 2nd highest priority and believing that only 1 copy should be pushed, is Passive. Server C sees that it is 3rd highest priority and believes 3 copies should be pushed, so it is Active. And server D sees it is 4th highest priority and, believing that only 2 copies should be pushed, is Passive.

If a Push Directory server is Active for Data Label X, it includes the Data Label X directory mappings it has in its ESADI-LSP for Data Label X and updates that information as the mappings it knows change. If the Push Directory server is configured to believe it has complete mapping information for Data Label X then, after it first actually transmits all of its ESADI-LSPs for X it then waits its CSNP time (see Section 6.1 of [ESADI]), and then updates its ESADI-Parameters APPsub-TLV to set the Complete Push (CP) bit to one. This change will cause its EASDI fragment zero to be flooded. It then maintains the CP bit as one as long as it is Active.

If a Push Directory server is Passive for Data Label X, it removes or continues to leave out all Data Label X directory mappings it holds from its ESADI-LSP for Data Label X. However, if it was Active and was advertising the CP bit as one in its ESADI-Parameters APPsub-TLV, it first updates the CP bit to zero and floods its updated ESADI-LSP fragment zero. Its then waits its CSNP time before withdrawing all its directory mapping information.

2.3 Additional Push Details

Push Directory mappings can be distinguished for any 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.

RBridges, whether or not they are a Push Directory server, MAY continue to advertise any locally learned MAC attachment information in [ESADI] 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 would generally not be done as it should not add anything and would just waste bandwidth and ESADI link state space. An exception would be when an RBridge 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 though it does not want to receive user data in those Data Labels, the No Data flag bit is provided as discussed in Section 6.3.

If an RBridge notices that a Push Directory server is no longer data reachable [RFCclear], it MUST ignore any Push Directory data from that server because it is no longer being updated and may be stale.

There may be transient 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 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.

3. Pull Model Directory Assistance Mechanisms

In the Pull Model, an RBridge pulls mapping information from an appropriate Directory Server when needed.

Pull Directory servers for a particular Data Label X are located by looking in the main TRILL IS-IS link state database for RBridges that advertise themselves by having the Pull Directory flag on in their Interested VLANs or Interested Labels sub-TLV [RFC6326bis] for X. If multiple RBridges indicate that they are Pull Directory Servers for a particular Data Label, a pull request can be sent to any of them that is data reachable but it is RECOMMENDED that pull requests be sent to server that is least cost from the requesting RBridge.

Pull Directory requests are sent by enclosing them in an RBridge Channel [Channel] message using the Pull Directory channel protocol number (see Section 6.2). Responses are returned in an RBridge Channel message using the same channel protocol number.

The requests to Pull Directory Servers are derived from normal ARP [RFC826], ND [RFC4861], RARP [RFC903] messages or data frames with unknown unicast destination MAC addresses intercepted by the RBridge when they would otherwise be ingressed. 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 for immediate response to ARP, ND, RARP, or unknown destination MAC frames, until they expire. If information previously pulled is about to expire, an RBridge MAY try to refresh it by issued a new pull request but, to avoid unnecessary requests, SHOULD NOT do so if it has not been recently used.

3.1 Pull Directory Request Format

A Pull Directory request is sent as the Channel Protocol specific content of an inter-RBridge Channel message TRILL Data packet. The Data Label in 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 request with the default mapping depending, per Data Label, on the strategy (see Section 4). The Channel Protocol specific data is formatted 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   V   | T |   RESV    | Count |              RESV             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| QUERY 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| QUERY 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| QUERY K
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

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

T: Type. 0 => Response, 1=> Query, 2=> Unsolicited Update, 3=> Reserved. An unsolicited update is formated as a response except there was no corresponding query. Messages received with type = 3 are discarded.

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

Count: Number of queries present.

Sequence Number: An opaque 32-bit quantity set by the sending RBridge, returned in any responses, and used to match up responses with queries.

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

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

SIZE: Size of the query data in bytes as an unsigned byt starting with and including the SIZE fiel itself. Thus the minimium legal value is 2. A value of SIZE less than indicates a malformed message. The "QUERY" with the illegal SIZE value and all subsequent QUERYs 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.

TYPE: There are two types of queries currently defines, (1) a query that provides an explicit address and asks for other addresses for the interface specified by the query address and (2) a query frame. The fields of each are specified below. Values of TYPE are as follows

TYPE	Description
0	reserved
1	query address
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: This is the query address. The query is asking for any other addresses that correspond to the same interface within the data label of the query and the RBridge from which they are reachable. Typically that would be either (1) a MAC address, in which case the querying RBridge is interested in the RBridge by which that MAC address is reachable, or (2) an IP address, in which case the querying RBridge is interested in the corresponding MAC address and the RBridge by which that MAC address is reachable.

Query Frame: Where a Pull Directroy query is the result of an ARP, ND, RARP, or unkown unicast MAC destination address, the ingress RBridge MAY send the frame to a Pull Directory Server if the frame is small enough to fit into a query message. This avoids the requirement that the ingress RBridge hold the frame pending a Pull Directory response.

A query count of zero is explicitly allowed, for the purpose of pinging a Pull Directory server to see if it is responding to

requests. On receipt of such an empty query messgea, a response message that also has a count of zero MUST be sent.

If no response is received to a Pull Directory request within a configurable timeout, the request should be re-transmitted with the same Sequence Number up to a configurable number of times that defaults to three. If there are multiple queries in a request, responses can be received to various subsets of these queries by the timeout. In that case, the remaining unanswered queries should be re-sent in a new query with a new sequence number. If an RBridge is not capable of handling partial responses to requests with multiple queries, it MUST NOT sent a request with more than one query in it.

3.2 Pull Directory Response Format

Pull Directory responses are sent as the Channel Protocol specific content of inter-RBridge Channel message TRILL Data packets. Responses are sent with the same Data Label and priority as the request to which they correspond except that the response priority is limited. This priority limit is configurable at a per RBridge level and defaults to priority 6. The 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   V   | T |F|P|N| RESV| Count |      ERR      |  subERR       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RESPONSE 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| RESPONSE 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| RESPONSE K
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

V: Version of the Pull Directory protocol. Version zero is specified in this document.

T: Type. 0 => Response, 1=> Query, 2=> Unsolicited Update, 3=> Reserved. An unsolicited update is formated as a response except there was no corresponding query. Unsolicited responses are sent to maintain cache consistency (see Section 3.5). Messages received with type = 3 are discarded.

F: The Flood bit. If zero, the reply is to be unicast to the provided Nickname. If T=2, F=1 is used to flood messages for certain unsolicited cache consistency maintenance messages from an end station Pull Directory server as discussed in Section 3.5. If T is not 2, F is ignored.

P, N: Flags used in connection with certain flooded unsolicited cache consistency maintenance messages. Ignored if T is not 2. If the P bit is a one, the solicited response message relates to cached positive response information. If the N bit is a one, the unsolicited messages related to cached negative information. See Section 3.5.

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

Count: Count is the number of responses present in the particular reponse message.

ERR, subERR: A two part error code. See Section 3.4.

Sequence Number: An opaque 32-bit quantity set by the requesting RBridge and copied by the Pull Directory into all responses to the query. For an unsolicited "response", the contents are unspecified.

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          |   RESV    |   Index   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                   Lifetime                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                Response Data ...
+--+--+--+--+--+--+--+--+--+--+--...
```

SIZE: Size of the response data in bytes starting with and including the SIZE field itself.

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

Index: The relative index of the query in the request message to which this response corresponds. The index will always be one for request messages containing a single query. The index will always be zero for unsolicited "response" messages.

Lifetime: The length of time for which the response should be

considered valid in seconds.

Response Data: There are two types of response data. If the ERR field is non-zero, the response data is a copy of the query data, that is, an AFN followed by an address. If the ERR field is zero, the response data is the contents of an Interface Addresses APPsub-TLV (see Section 5) without the usual TRILL GENINFO TLV type and length and without the usual IA APPsub-TLV type and length before it.

Multiple response records can appear in a response message with the same index if the answer to a query 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 RBridges.

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.

See Section 3.4 for a discussion of how errors are handled.

3.3 Pull Directory Hosted on an End Station

Optionally, a Pull Directory actually hosted on an end station MAY be supported. In that case, when the RBridge advertising itself as a Pull Directory server receives a query, 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 RBridge of the query.

The native RBridge Channel Pull Directory messages use the same Channel protocol number as do the inter-RBridge Pull Directory Channel messages. The native messages MUST 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.

The native RBridge Channel message protocol dependent data for a Pull Directory query is formatted 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   V   | T |   RESV    | Count |           Nickname            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Data Label ... (4 or 8 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| QUERY 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| QUERY 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| QUERY K
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

Data Label: The Data Label of the original inter-RBridge Pull Directory Channel protocol messages that was mapped to this native channel message. The format is the same as it appears right after the Inner.MacSA of the original Channel message.

Nickname: The nickname of the requesting RBridge.

All other fields are as specified in Section 3.1.

The native RBridge Channel message protocol specific content for a Pull Directory response is formatted 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   V   | T |F|P|N| RESV| Count |      ERR      |  subERR       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Nickname            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Data Label ... (4 or 8 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RESPONSE 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| RESPONSE 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| RESPONSE K
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

Data Label: The Data Label to which the response applies. The format is the same as it appears right after the Inner.MacSA in TRILL Data messages.

Nickname: The nickname of the destination RBridge or, if F=1, ignored.

All other fields are as specified in Section 3.2.

3.4 Pull Directory Request Errors

An error response message is indicated by a non-zero ERR field.

If there is an error that applies to the entire request 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 expanded with a zero Lifetime and the insertion of the Index field echoed back in the response records.

If errors occur at the query level, they MUST be reported in a response message separate from the results of any successful queries. If multiple queries in a request have different errors, they MUST be reported in separate response messages. If multiple queries in a request have the same error, this error response MAY be reported in one response message.

In an error response message, the query or queries 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.

ERR values 1 through 63 are available for encoding request message level errors. ERR values 64 through 255 are available for encoding query 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 V field value
2	Request data too short
3	Administratively prohibited
4-31	(Available for allocation by Standards Action)
32	Unknown AFN
33	No mapping found
34	Administratively prohibited
35-255	(Available for allocation by Standards Action)

More TBD...?

3.5 Cache Consistency

Pull Directories MUST take action to minimize the amount of time that an RBridge will continue to use stale information from the Pull Directory.

A Pull Directory server MUST maintain one of the following, in order of increasing specificity.

1. An overall record per Data Label of when the last returned query data will expire at a requestor and when the last query record specific negative response will expire.

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

3. For each unit of data held by the server and each address about which a negative response was sent, a list of RBridges that were sent that unit as the response or sent a negative response to the address, with the expected time to expiration at each of them.

A Pull Directory server may have a limit as to how many RBridges it can maintain expiry information for 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 querying RBridge, the Pull Directory MUST send an unsolicited message as discussed below.

If method 1, the most crude method, is being followed, then when any information in a Data Label is changed or deleted or an additional administrative Pull Directory access restriction imposed, and there are outstanding cached positive query data response(s), an all-addresses flush positive message is flooded (multicast) within that Data Label. And if data is added or an administrative restriction is removed and there are outstanding cached negative responses, an all-addresses flush negative message is flooded. "All-addresses" is indicated by the Count in an unsolicited response being zero. On receiving an all-addresses flooded flush positive message from a Pull Directory server it has used, indicated by the U, F, and P bits being one, an RBridge discards all cached data responses it has for that Data Label. Similarly, on receiving an all addresses flush negative message, indicated by the U, F, and N bits being one, it discards all cached negative responses for that Data Label. A combined flush positive and negative can be flooded by having all of the U, 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 an RBridge floods address specific update positive unsolicited responses when data which is cached by a querying RBridge is changed or deleted or an administrative restriction is added to such data and floods an address specific update negative unsolicited responses when such information is deleted or an administrative restriction is removed from such data. Such messages are similar to the method 1 flooded unsolicited flush messages. The U and F bits will be one and the message will be multicast. However that 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 message, if it is positive the addresses in the response records in the unsolicited response are compared to the addresses about which the recipient RBridge is holding cached positive information and, if they match, the cached information is updated and its remaining cache life set to the minimum of its previous value in the cache and the Lifetime value in the unsolicited response. In the case of a newly imposed administrative restriction, the Lifetime in the unsolicited response is set to zero so the cached information immediately expired. On receiving an address specific unsolicited negative response, the addresses in the response records in the unsolicited response are compared to the addresses about which the recipient RBridge is holding cached negative information and, if they match, the cached negative information is discarded.

If method 3 is being followed, the same sort of messages are sent as with method 2 except they are not flooded but unicast only to the specific RBridges the server believes may be holding the cached

positive or negative information that may need updating.

3.6 Additional Pull Details

If an RBridge notices that a Pull Directory server is no longer data reachable [RFCclear], it MUST discard all responses it is retaining from that server within one second as the RBridge can no longer receive cache consistency messages from the server.

Because a Pull Directory server may need to advertise interest in Data Labels even though it does not want to received user data in those Data Labels, the No Data flag bit is provided as discussed in Section 7.3.

4. Directory Use Strategies and Push-Pull Hybrids

For some edge nodes which have great number of Data Labels enabled, managing the MAC&Label <-> RBridgeEdge 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 RBridge Edge 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 when hosts in specific Data Labels, say VLANs 1 through 100, communicate regularly with external peers, 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 RBridges 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.

4.1 Strategy Configuration

Each RBridge 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 RBridge 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.

1. Use Push only. The RBridge subscribes to Push Directory data for Data Label X.

2. Use Pull only. When the RBridge 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 RBridge 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 modes 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 network connectivity and so is avoided by default for directory traffic.

5. Security Considerations

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

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

For general TRILL security considerations, see [RFC6325].

6. IANA Considerations

This section give IANA allocation and registry considerations.

6.1 ESADI-Parameter Data

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

Sub-Registry: ESADI-Parameter APPsub-TLV Bits

Registration Procedures: IETF Review

References: [ESADI], This document

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

In addition, the ESADI-Parameter APPsub-TLV is optionally extended, as provided in its original specification in [ESADI], 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 [ESADI] 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 place here by an

RBridge for which this field has not been configured.

6.2 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 table of such protocol number in the TRILL Parameters Registry referencing this document.

6.3 Pull Directory and No Data Bits

IANA is requested to allocate two currently reserved bits in the Interested VLANs field of the Interested VLANs sub-TLV (suggested bits 3 and 4) and the Interested Labels field of the Interested Labels sub-TLV (suggested bits 5 and 6) [RFC6326bis] to indicate Pull Directory server (PD) and No Data (ND) respectively. These bits are to be added to the subregistry set up in [ESADI].

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 an RBridge indicates connection to end stations in the VLANs or FGLs 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 as part for the Data Label you are serving. If such a directory server does not wish to received multi-destination user data for the Data Labels it lists in one of these sub-TLVs, it sets the "No Data" (ND) bit to one. This means that data on a distribution tree may be pruned so as not to reach the "No Data" RBridge as long as there are no RBridges interested in the Data who are beyond the "No Data" RBridge. This bit is backwards compatible as RBridges ignorant of it will simply no prune when it could, which is safe but may cause increased link utilization.

Acknowledgments

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

[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.

[RFC5342bis] - Eastlake 3rd, D., "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters", BCP 141, RFC 5342, September 2008.

[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-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.

[ESADI] - 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-appsub-tlv, work in progress.

Informational References

[DirectoryFramework] - Dunbar, L., D. Eastlkae, R. Perlman, I. Gashinsky, "TRILL Edge Directory Assistance Framework", draft-ietf-trill-directory-framework, 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: (469) 277 5840
Email: ldunbar@huawei.com

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

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

Radia Perlman
Intel Labs
2200 Mission College Blvd.
Santa Clara, CA 95054-1549 USA

Phone: +1-408-765-8080
Email: Radia@alum.mit.edu

Igor Gashinsky
Yahoo
45 West 18th Street 6th floor
New York, NY 10011

Email: igor@yahoo-inc.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012 China

Phone: +86-25-56622310
Email: liyizhou@huawei.com

Copyright, Disclaimer, and Additional IPR Provisions

INTERNET-DRAFT
Intended status: Proposed Standard

Donald Eastlake
Yizhou Li
Huawei
Radia Perlman
Intel

Expires: January 14, 2014
July 15, 2013

TRILL: Interface Addresses APPsub-TLV
<draft-eastlake-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 sets of addresses such that all of the addresses in each set designate the same interface (port). For example, an EUI-48 MAC (Extended Unique Identifier 48-bit, Media Access Control) address, IPv4 address, and IPv6 address can be reported as all corresponding to the same interface. Such information could be used, for example, to synthesize responses to or by-pass the need for the Address Resolution Protocol (ARP), the IPv6 Neighbor Discovery (ND) protocol, of the flooding of unknown MAC addresses, in some cases.

Status of This Memo

Table of Contents

1. Introduction

This document specifies a TRILL (Transparent Interconnection of Lots of Links) [RFC6325] IS-IS application sub-TLV (APPsub-TLV [RFC6823]) that enables the convenient representation of sets of addresses such that all of the addresses in each set designate the same end station interface (port). For example, an EUI-48 MAC (Extended Unique Identifier 48-bit, Media Access Control [RFC5342bis]) address, IPv4 address, and IPv6 address can be reported as all three corresponding to the same interface.

This APPsub-TLV is used inside the TRILL GENINFO TLV as specified in [ESADI]. It is expected to be used in Directory Assisted TRILL Edge services [DirectoryFramework].

Although, in some IETF protocols, address field types are represented by EtherType [RFC5342bis] or Hardware Type [RFC5494] only Address Family Number is used in this APPsub-TLV.

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].

2. Format of the Interface Addresses APPsub-TLV

The Interface Addresses APPsub-TLV is used to indicate that a set of addresses indicate the same end-station interface and to associate that interface with the TRILL switch by which the interface is reachable. These addresses can be in different address families. For example, it can be used to declare that an end-station interface with a particular IPv4 address, IPv6 address, and EUI-48 MAC address is reachable from a particular TRILL switch.

The Template field indicates certain well known sets of addresses or gives a number of AFNs. When AFNs are listed, the set of AFNs provides an explicit template for the type and order of addresses in each Address Set.

```
+-+-+-+-+-+-+-+-+
| Type = TBD    |                  (1 byte)
+-+-+-+-+-+-+-+-+
| Length        |                  (1 byte)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Nickname                      |  (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Flags         |                  (1 byte)
+-+-+-+-+-+-+-+-+
| Confidence    |                  (1 byte)
+-+-+-+-+-+-+-+-+
| Addr Set End  |                  (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 5. If length is 4 or less, the APPsub-TLV MUST be ignored.

- o Nickname: The nickname of the RBridge by which the address sets

are reachable.

o Flags: A byte of flags as follows:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|D|L|   Resv    |
+-+-+-+-+-+-+-+-+
```

D: If D is one, the APPsub-TLV contains Push Directory information.

L: If L is one, the APPsub-TLV contains information learned locally be observing ingressed frames. (Both D and L can one in the same APPsub-TLV.)

Resv: Additional reserved flag bits that MUST be sent as zero and ignored on receipt.

o Confidence: This 8-bit quantity indicates the confidence level in the addresses being transported [RFC6325].

o Addr Set End: The unsigned offset of the byte, within the TLV value part, of the last byte of the last Address Set. This will be the byte just before the first sub-TLV if any sub-TLVs are present. [RFC5305]

o Template: The initial byte of this field is the unsigned integer K. It K has a value from 1 to 63, it indicates that this initial byte is followed by a list of K AFNs (Address Family Numbers) in the template specifying the structure and order of each Address Set occurring later in the TLV. The minimum valid value is 1. If K is 64 to 255, it indicates that the Template for each Address Set is a specific well known Template. If the Template includes explicit AFNs, they look like the following.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AFN 1                         |  (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AFN 2                         |  (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AFN K                         |  (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

For K in the 64 to 255 range, some values indicate combinations of a specific number of 48-bit MAC addresses, IPv4 addresses, and IPv6 addresses in that order. If M is the number of MAC addresses (limited to 1 or 2), 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 4 inclusive), the value of K is

K = 63 + M + 2*v4 + 6*v6

That equation specifies values of K from 64 through 93. Values from 94 through 255 are available for assignment by IETF Review.

- o AFN: A two-byte Address Family Number. The number of AFNs present is given in first byte of the Template field if that value is less than 64. This 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 EUI-48 and IPv4, 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-TLV as specified below, the receiving IS will be unable to parse the Address Sets and MUST ignore the enclosing TLV.

- o Address Set: Each address set consists of a sequence of addresses of the types given by the Template earlier in the 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 Address Sets. The Address Sets must fit within the TLV. If the product of the size of an Address Set and the number of Address Sets is so large that this is not true, the APPsub-TLV is ignored.

- o sub-sub-TLVs: If the Address Sets indicated by Addr Sets End do not completely fill the Length of the TLV, the remaining bytes are parsed as sub-sub-TLVs [RFC5305]. Any such sub-sub-TLVs that are not known to the receiving RBridge are ignored. Should this 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 ESADI-LSPs or Pull Directory responses from the same RBridge may have different Templates. The same AFN may occur more than once in a Template and the same address may occur in more than one address set. For example, an EUI-48 MAC address interface might have three 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 EUI-48 and one IPv6 address could be used in an IA-APPsub-TLV with three address sets each having the same EUI-48 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 RBridge must know at least the size of the address each AFN in the Template specifies; however, the presence of the Addr Set End field means that the sub-TLVs, if any, can always be located by a receiving IS. An RBridge can be assumed to know the size of EUI-48, IPv4, and IPv6 addresses (AFNs 16389, 1, and 2) and the size of the additional AFNs allocated by the IANA Considerations below. Should an RBridge wish to include an AFN that some receiving RBridge in the campus may not know, it SHOULD include an AFN-Size sub-sub-TLV as described below. If an IA-APPsub-TLV is received with one or more AFNs in its template for which the receiving RBridge does not know the length and for which an AFN-Size sub-sub-TLV is not present, that IA-APPsub-TLV will be ignored.

3. IA-APPsub-TLV sub-sub-TLVs

IA-APPsub-TLVs may 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.

3.1 AFN Size sub-sub-TLV

Using this sub-TLV, the originating RBridge can specify the size of an address type. This is useful under two circumstances:

1. One or more AFNs that are unknown to the receiving RBridge 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.

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-APPsubTLV.

```
+-+-+-+-+-+-+-+-+
| Type = AFNsz  |                  (1 byte)
+-+-+-+-+-+-+-+-+
| Length        |                  (1 byte)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AFN Size Record(s)                            |  (3 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where each AFN Size Record is structured as follows:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AFN                          |  (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AdrSize      |                  (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 n 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 AdrSize: The length of the address specified by the AFN field.

This sub-sub-TLV may occur multiple times in an enclosing IA-APPsub-TLV.

An AFN Size sub-sub-TLV for any AFN known to the receiving RBridge (which always includes AFN 1, 2, and 16389 and the AFNs specified in xxx) is compared with the size known to the RBridge and if they differ, the IA-APPsub-TLV is ignored.

3.2 Fixed Address sub-sub-TLV

There may be cases where, in an Interface Addresses TLV, the same address would appear across every address set in the TLV. To avoid having a larger template and wasted space in all Address Sets, this sub-sub-TLV can be used to indicate such a fixed address

```
+-+-+-+-+-+-+-+-+
|Type=FIXEDADR  |                 (1 byte)
+-+-+-+-+-+-+-+-+
| Length        |                 (1 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.

3.3 Data Label sub-sub-TLV

When used with Push or Pull Directories, the Data Label is indicated by the Data Label of the ESADI instance (Push) or RBridge Channel message (Pull) in which the IA APPsub-TLV appears and any occurrence

of this sub-sub-TLV is ignored. However, the IA APPsub-TLV might be used in other contexts where this sub-sub-TLV indicates the Data Label of the Address Sets and multiple occurrences of this sub-sub-TLV indicate that the Address Sets exist in all of the Data Labels.

```
+-+-+-+-+-+-+-+-+
|Type=DATALEN   |                 (1 byte)
+-+-+-+-+-+-+-+-+
| Length        |                 (1 byte)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
| Data Label                      (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

o Type: Data Label sub-TLV type, set to 3 (DATALEN).

o Length: 2 or 3

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 Address Sets are in the topology give. If it occurs multiple times, then the Address Sets are in all of the topologies listed.

```
+-+-+-+-+-+-+-+-+
|Type=DATALEN   |                  (1 byte)
+-+-+-+-+-+-+-+-+
| Length        |                  (1 byte)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RESV  |        Topology       |  (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

o Type: Data Label sub-TLV type, set to 3 (DATALEN).

o Length: 2.

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

o Topology: The 12-bit topology number [RFC5120].

4. Security Considerations

TBD...

5. IANA Considerations

5.1 Additional AFN Number Allocation

IANA is requested to allocate three new AFN numbers as follows:

Number	Description	References
TBD(29)	OUI	[RFC5342bis], this document
TBD(30)	MAC/24	This document.
TBD(31)	IPv6/64	This document.

The OUI AFN is provided so that MAC addresses can be abbreviated if they have the same upper 24 bits. In particular, if there is an OUI provided as a Fixed Address sub-sub-TLV (see Section 5.2.2) then, whenever a MAC/24 address appears within an Address Set (as indicated by the Template), the OUI is used as the first 24 bits of the actual MAC address for the Address Set.

MAC/24 is a 24-bit suffix intended to be pre-fixed by an OUI as in the previous paragraph. In absence of an OUI specified as a Fixed Address in the same APPsub-TLV, an Address Set containing an MAC/24 address cannot be used.

IPv6/64 is an 8-byte quantity that is the first 64 bits of an IPv6 address. If present, there will normally be an EUI-48 or EUI-64 address in the address set to provide the lower 64 bits of the IPv6 address. For this purpose, an EUI-48 is expanded to 64 bits as described in [RFC5342bis].

5.2 IA APPsub-TLV Sub-Sub-TLVs SubRegistry

IANA is requested to establish a new subregistry 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: IETF Review

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	This document
255	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.

Normative References

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

[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.

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

[RFC5342bis] - Eastlake 3rd, D., "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters", BCP 141, RFC 5342, September 2008.

[RFC5494] - Arkko, J. and C. Pignataro, "IANA Allocation Guidelines for the Address Resolution Protocol (ARP)", RFC 5494, April 2009.

[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.

[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.

[ESADI] - 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.

Informational References

[DirectoryFramework] - Dunbar, L., D. Eastlkae, R. Perlman, I. Gashinsky, "TRILL Edge Directory Assistance Framework", draft-ietf-trill-directory-framework, work in progress.

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

Authors' Addresses

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

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

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012 China

Phone: +86-25-56622310
Email: liyizhou@huawei.com

Radia Perlman
Intel Labs
2200 Mission College Blvd.
Santa Clara, CA 95054-1549 USA

Phone: +1-408-765-8080
Email: Radia@alum.mit.edu

Copyright, Disclaimer, and Additional IPR Provisions

TRILL Working group
Internet Draft
Intended status: Standard Track

Tissa Senevirathne
Norman Finn
Samer Salam
Deepak Kumar
CISCO

Donald Eastlake
Sam Aldrin
Yizhou Li
Huawei

July 10, 2013

Expires: January 2014

TRILL Fault Management
draft-ietf-trill-oam-fm-00.txt

Status of this Memo

Copyright Notice

Abstract

This document specifies TRILL OAM Fault Management. Methods in this document follow the IEEE 802.1 CFM 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.

Table of Contents

1. Introduction

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

The OAM message channel allows defining various control information and carrying 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.

This document uses the message format defined in IEEE 802.1ag Connectivity Fault Management (CFM) [8021Q] as the basis for the TRILL OAM message channel.

The ITU-T Y.1731 [Y1731] standard utilizes the same messaging format as [8021Q] and OAM messages where applicable. This document takes a similar stance and reuse [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.

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:

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]

CCM - Continuity Check Message [8021Q]

LBM - Loop Back Message [8021Q]

PTM - Path Trace Message

MTV - Multi-destination Tree Verification Message

OAM - Operations, Administration, and Maintenance [RFC6291]

TRILL - Transparent Interconnection of Lots of Links [RFC6325]

FGL - Fine Grained Label [RFCfgl]

ECMP - Equal Cost Multipath

ISS - Internal Sub Layer Service [8021Q]

3. General Format of TRILL OAM frames

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.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
.    Link  Header               . (variable)
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+    TRILL Header               + 8 or more bytes
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
.   Flow Entropy                . 96 bytes
.                               .
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   OAM Ether Type              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
.   OAM Message Channel         . Variable
.                               .
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Link Trailer              | Variable
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 least significant 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].

OAM Ether Type: OAM Ether Type is 16-bit EtherType that identifies the OAM Message channel that follows. This document

specifies using the EtherType allocated for 802.1ag for this purpose. Identifying the OAM Message Channel with a dedicated EtherType allows the easy identification of the beginning of the OAM message channel across multiple standards.

OAM Message Channel: This is a variable size section that carries OAM related information. The message format defined in [8021Q] will be reused for TRILL OAM.

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 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".

Implementations that comply with this document MUST utilize "A" flag and CFM etherType to identify TRILL OAM frames. The "A" flag MUST NOT BE utilized for forwarding decisions such as the selection of which ECMP path or multi-destination tree to use.

```
                                    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   +-+
                                    | 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 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 TRILL 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 Flags are defined:

O - OAM Capable

B - Backwards Compatible.

A capability announcement, with O Flag set to 1 and B flag set to 1, indicates that the implementation is OAM capable but utilize 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 implementation 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 in this case the B flag is ignored.

```
+-+-+-+-+-+-+-+-+
| Type          |              (1 byte)
+-+-+-+-+-+-+-+-+
| Length        |              (1 byte)
+-+-+-+-+-+-+-+-+
| Max-version   |              (1 byte)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+
|A|O|B|Other Capabilities and Header Flags|  (4 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+-+
 0                   1                 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7   0 1
```

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

NOTE: Bit position of O and B flags in the TRILL-VER sub-TLV are presented above as an example. Actual positions of the flags will be determined by TRILL WG and IANA and future revision of this document will be updated to include the allocations.

4. TRILL OAM Layering vs. IEEE Layering

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

```
                        +-+-+-+-+-+-+-+-+-+-+
                        |   RBridge Layer   |
                        |   Processing      |
                        +-+-+-+-+-+-+-+-+-+-+
                                 |
                                 |
                             +-+-+-+-+-+-+
                             | TRILL OAM | UP MEP
                             | Layer     |   MIP
                             +-+-+-+-+-+-+ Down MEP
                                  |
                                  |
                             +-+-+-+-+-+-+
      (3)-------->           | TRILL     |
                             | Encap/Decap
                             +-+-+-+-+-+-+
                                 |
                             +-+-+-+-+-+-+
      (2)-------->           |End station|
                             | VLAN & priority Processing
                             +-+-+-+-+-+-+
                                 |
                             +-+-+-+-+-+-+
      (1)-------->           |ISS        |
                             |Processing |
                             +-+-+-+-+-+-+
                                 |
                                 |
                                 |
```

Figure 4 Placement of TRILL MP within IEEE 802.1

[RFC6325] Section 4.6 as updated by [RFCclcorrect] provides a detailed explanation of frame processing. Please refer to [RFC6325] 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, our Outer.VLAN because these are part of the link header for Ethernet. Point-to-point links typically have link headers

without these fields. These fields are primarily significant for native frames from and/or to end stations.

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 an 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 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)

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 OAM 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.VL 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.VL 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 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.

- 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

- 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 Received 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 Functionality that 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 802.1ag MIB definitions, each MEP’s MIB contains one or more flow entropy definitions corresponding to the set of flows that the MEP monitors.

[8021Q] 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.

```
             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)
           |
            Other MIB entries
```

Figure 5 Correlation of TRILL augmented MIB

6. MEP Addressing

In IEEE 802.1ag [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.

```
                                |
                                .CCM   LBM   PTM    MTV
                                |      |    |      |
                              +-+-+-+-+-+-+-+-+-+-+-+-+
                              |        OP Code DE-Mux |--- Unknown
                              +-+-+-+-+-+-+-+-+-+-+-+-+
                                ^       ^          ^
                       MD==Li   |       |          |
                              +-+-+-   +-+-+      +-+-+
                              | L |-->|L2 |-.-   |Ln |---- >
                              +-+-+-   +-+-+      +-+-+      |
                               |  ^    |          |         |
                       MD<LI Drop |    Drop       Drop      |
                                  |                         |
                       MD not --- |TRILL OAM need local     |
                       Present    | Processing              |
                                  |                         |
                    TRILL Data   ----  TRILL Data         ----
                       ------->| T  |---------------- >|  M |--- >
                    + TRILL OAM  ----  + pass through OAM ----
```

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 OAM flag set AND OAM Ether type 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.

Similar approach as 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.

```
                                                PT     MTV
                                                 |      |
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |             OP Code De-Mux  |->
   Unknown
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                  ^       ^          ^
                       MD==Li     |       |          |
                              +-+-+   +-+-+      +-+-+
                              | L |- >|L2 |-.-   |Ln |------+
                              +-+-+   +-+-+      +-+-+      |
                                  ^                         |
                                  |                         |
                       Drop       |                         |
                       MD not --- |TRILL OAM                |
                       Present    |                         |
                                  |                         v
                     TRILL Data   ----  TRILL Data          -----
                        ------- >| T  |------------------ >|  M  |----
   >
                     + TRILL OAM  ----                      ----
```

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) {note we have to define
                   |                      destination nickname with
                    --- (Flow-entropy-2)  the flow entropy discuss}
                   |
                   . ---(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 can be accomplished by re-using the OAM message header defined in [8021Q].

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|MD-L | Version | OpCode        |  Flags        |FirstTLVOffset |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.   Opcode Specific Information                                 .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.         TLVs                                                  .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 zero. 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 OAM Opcodes

The following Opcodes are defined for TRILL. 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:

```
 TBD-64 : Path Trace Reply
 TBD-65 : Path Trace Message
 TBD-66 : Multicast Tree Verification Reply
 TBD-67 : Multicast Tree Verification Message
```

8.3. Format of TRILL OAM TLV

The same TLV format as defined in section 21.5.1 of [8021Q] is used for TRILL OAM. The following figure depicts the general format of a TRILL OAM TLV:

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type       |        Length                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                               |
.            Value(variable)                    .
|                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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. Types 32-63 are reserved for ITU-T Y.1731. We propose to reserve Types 64-95 for TRILL OAM TLVs.

8.4.1. Common TLVs between 802.1ag and TRILL

The following TLVs are defined in [8021Q]. We propose to 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

As indicated above, Types 64-95 will be requested to be reserved for TRILL OAM purposes. 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
TBD-TLV-64	TRILL OAM Application Identifier
TBD-TLV-65	Out of Band IP Address
TBD-TLV-66	Original Payload
TBD-TLV-67	Diagnostic VLAN
TBD-TLV-68	RBridge scope
TBD-TLV-69	Previous RBridge Nickname
TBD-TLV-70	TRILL Next Hop RBridge List (ECMP)
TBD-TLV-71	Multicast Receiver Availability
TBD-TLV-72	Flow Identifier
TBD-TLV-73	Reflector Entropy
TBD-TLV-74 to TBD-TLV-95	Reserved

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 an RBridge, unless that RBridge is also running Ethernet CFM.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type       | Length                        | Version       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Return Code  |Return sub-code|      Reserved         |F|C|O|I|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11 TRILL OAM Application Identifier TLV

Type (1 octet) = TBD-TLV-64 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.

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.

```
                      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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type       | Length                        | Address Type  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Addr Length   |                                               |
+-+-+-+-+-+-+-+-+                                               |
|                                                               |
.       Reply Address                                           .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12 Out of Band IP Address TLV

Type (1 octet) = TBD-TLV-65

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.

```
                        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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+
    |    Type       | Length                        | L-Type
|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+
    | Reserved      |                       Label(VLAN)
|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+
```

Figure 13 Diagnostic VLAN TLV

Type (1 octet) = TBD-TLV-66 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 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

```
                        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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Type       | Length                        |               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +
   |                                                               |
   .                Original Payload                               .
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14 Original Data Payload TLV

Type (1 Octet) = TBD-TLV-67

Length (2 octets) = variable

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.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type       | Length                        | nOfnicknames  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  nickname-1                   |   nickname-2                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |  nickname-n                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15 RBridge Scope TLV

Type (1 octet) = TBD-TLV-68 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).

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type       | Length                        | Reserved      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Reserved                     |   nickname                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16 Previous RBridge nickname TLV

Type (1 octet) = TBD-TLV-69 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.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type       | Length                        | nOfnicknames  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  nickname-1                   |   nickname-2                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |  nickname-n                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17 Next Hop RBridge List TLV

Type (1 octet) = TBD-TLV-70 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.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type       | Length                        | Reserved      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              number of Receivers                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18 Multicast Receiver Availability TLV

Type (1 octet) = TBD-TLV-71 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.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type       | Length                        | Reserved      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  MEP-ID                       |     flow-id                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19 Flow Identifier TLV

Type (1 octet) = TBD-TLV-72

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 inclusion of MEP-ID for messages that do not contain MEP-ID in 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.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type       | Length                        | Reserved      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.               Reflector Entropy                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20 Reflector Entropy TLV

Type (1 octet) =TBD-TLV-73 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.

9. Loopback Message

9.1. Loopback OAM Message format

```
                      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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|MD-L | Version | OpCode        |  Flags        |FirstTLVOffset |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Loopback Transaction Identifier             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.         TLVs                                                  .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 65 and the Opcode for the Reply Message is set to 64. 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

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). 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 and 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 includes it in the OAM message.

Set the Return Code and Return sub code to applicable values. 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, TRILL is multicast and unicast incongruent. Hence, TRILL OAM uses a new message format: the Path Trace message.

The Path Trace Message has the same format as Loopback Message. Opcode for Path Trace Reply Message is 65 and Request 64

Operation of the Path Trace message is identical to the Loopback message except that it is first transmitted with a TRILL 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 (65). 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 OAM 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 (2) "Time Expired" and Return sub code to zero (0). Update the TRILL OAM opcode to 64 (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 (64).

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 RBridges 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.

11.2. Theory of Operation

11.2.1. Actions by Originator RBridge

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 needed 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 RBridge 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 to map the response to the correct request.

TRILL OAM Application Identifier TLV MUST be included.

Op-Code MUST be specified as Multicast Tree Verification Message (70)

Include RBridge scope TLV (67)

Optionally, include following TLV, where applicable

- o Out-of-band IP address
- o Diagnostic Label
- 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.

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 66 (Multicast Tree Verification Reply).

Include following TLVs:

Upstream RBridge nickname TLV (69)

Reply Ingress TLV (5)

Interface Status TLV (4)

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

Sender ID TLV (1)

Multicast Receiver Availability TLV (71)

If Label (VLAN or FGL) cross connect error detected, set 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. 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 (1) 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 TRILL OAM TLVs, where applicable

- 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 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

The response Message allows Fragmented Replies. In case of Fragmented Replies, all messages MUST follow the procedure defined in this section.

All Reply Messages MUST be encoded as described in this document.

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 the appropriate Final Flag field. The Final Flag, MUST, only be set on the final fragment of the reply.

14. Security Considerations

For general TRILL related security considerations, please refer to [RFC6325]. Specific security considerations related methods presented in this document are currently under investigation.

15. IEEE Allocation Considerations

The IEEE 802.1 Working Group is requested to allocate a separate opcode and TLV space within 802.1QCFM messages for TRILL purpose.

16. IANA Considerations

IANA is requested to do the following:

- Assign a multicast MAC address from the block assigned to TRILL [RFC6325]

- Set up sub-registry within the TRILL Parameters registry for block of TRILL "OAM OpCodes" (Section 8.2.)

- Set up sub-registry within the TRILL Parameters registry for TRILL "OAM TLV Types" (Section 8.4.)

- Assign a unicast MAC addressed under the IANA OUI, reserved for identification of OAM packets discussed in backward compatibility method (Appendix A) See Appendix C.

17. References

17.1. Normative References

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

[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.

17.2. 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 f<or 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 201.

[RFC6905] Senevirathne, T. et.al"Requirements for Operations, Administration, and Maintenance (OAM) in Transparent Interconnection of Lots of Links (TRILL)", RFC 6905, March 2013.

[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.

[TRILLEXT] Eastlake, Donald, et.al. "TRILL: Header Extension", draft-ietf-trill-rbridge-extension, June, 2012.

[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 Edtior’s queue.

[TRILLOAMMIB] Deepak Kumar et.al, "TRILL OAM MIB", draft-deepak-trill-oam-mib, May 2013.

18. 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 RBridges, 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 RBridge 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, RBridges MUST have the capability to prevent the leaking of OAM packets, as specified in [RFC6905].

[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 will 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.

A.2 Data plane encoding and frame identification

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 OAM 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.

Identification of Unicast non-IP OAM Flows,

Identification of Multicast non-IP OAM Flows,

Identification of Unicast IP OAM Flows and

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 source MAC field or 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 RBRdige nickname of the packet MUST match that of the local RBRidge or for multicast flows, TRILL header mutlicast 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

It is important to note that all RBridges MUST generate OAM flows with "A" flag set and CFM EttherType "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 RBRdiges that are compliant to the backwards compatibility method defined in this document.

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, per the Operational and Manageability considerations described in Section 6 of [TRLOAMFRM].

All RBrdiges 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.

In the Base Mode of operation, each RBrdige 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.

Implementation 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    | 1       |
|Domain Format  |         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Maintenance    | 2       |
|Domain Length  |         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Maintenance    | variable|
|Domain Name    |         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Short MA       | 1       |
|Name   Format  |         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Short MA       | 2       |
|Name  Length   |         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Short MA       | variable|
|Name           |         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|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

Shirt 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: 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: Unicast

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: Standards Track
Expires: January 2014

T. Mizrahi
Marvell
T. Senevirathne
S. Salam
D. Kumar
Cisco
D. Eastlake 3rd
Huawei

July 14, 2013

Loss and Delay Measurement in Transparent Interconnection of Lots of Links (TRILL)

draft-mizrahi-trill-loss-delay-01.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 (LM) and Delay Measurement (DM) in TRILL networks.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

TRILL [RFCTRILL] is a protocol for transparent least cost routing, where RBridges forward traffic to their destination based on a least cost route, 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 (LM) and Delay Measurement (DM). PM may also include other derived metrics such as Packet Delivery Rate (PDR), and Inter-Frame Delay Variation (IFDV).

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 are somewhat based on the mechanisms defined in ITU-T G.8013/Y.1731 [Y.1731].

- o Loss Measurement (LM): the LM 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 loss rate 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 (DM): the DM 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 that implements TRILL PM MUST/SHOULD/MAY/...’.

2.2. Definitions

- o One-way packet delay - (as defined in [OAM-REQ]) 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 - (as defined in [OAM-REQ]) 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 - the number of packets lost in a specific probe instance, and a specific observation period.

- 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
1LM	One-way Loss Measurement message
DM	Delay Measurement
DMM	Delay Measurement Message
DMR	Delay Measurement Reply
FD	Frame Delay
FDR	Frame Delay Range
FLR	Frame Loss Ratio
IFDV	Inter-Frame Delay Variation
MD	Maintenance Domain
MD-L	Maintenance Domain Level
MEP	Maintenance End Point
MFD	Mean Frame Delay
MIP	Maintenance Intermediate Point
MP	Maintenance Point
LM	Loss Measurement
OAM	Operations, Administration and Maintenance
OWDM	One-Way Delay Measurement
OWLM	One-Way Loss Measurement

PDR Packet Delivery Rate

PM Performance Monitoring

TLV Type, Length and Value

TRILL Transparent Interconnection of Lots of Links

TWDM Two-Way Delay Measurement

TWLM Two-Way Loss Measurement

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).

```
+-------+     +-------+     +-------+
|       |     |       |     |       |
|  RB1  |<===>|  RB3  |<===>|  RB2  |
|       |     |       |     |       |
+-------+     +-------+     +-------+
   MEP           MIP           MEP
```

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, i.e., 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. Therefore, this document provides tools for one-way performance monitoring and for two-way performance monitoring.

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 functionality of the sender, RB1, and the 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 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 PM 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.

4. Loss Measurement

The LM protocol has two flavors, One-Way Loss Measurement (OWLM), and Two-Way Loss Measurement (TWLM).

Notes: [Y.1731] defines two-way LM, but does not support one-way LM. The terms ’one-way’ and ’two-way’ LM should not be confused with the terms ’single-ended’ and ’dual-ended’ LM 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 (OWLM)

OWLM measures the one-way packet loss rate from one MEP to another. The loss rate 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 an OWLM message exchange.

```
                     TXp              TXc
        Sender    -------------------------------------
                        \                 \
                         \ 1SL   . . .    \ 1SL
                          \                \
                          \/               \/
        Receiver  -------------------------------------
                          RXp              RXc
```

Figure 2 One-Way Loss Measurement

The OWLM procedure uses a set of 1SL messages to measure the packet loss rate. 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 rate by comparing its counter values to the counter values received in the 1SL messages.

In Figure 2, the subscript ’c’ is short for current, and ’p’ is short for previous.

4.1.1. 1SL Message Transmission

OWLM 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 OWLM 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 OWLM MUST support unicast transmission of 1SL messages.

A MEP that supports OWLM 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. Therefore, the first packet of a probe instance is sent with the counter value set to 1.

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 OWLM 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):

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

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.

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.

4.2. Two-Way Loss Measurement (TWLM)

TWLM allows a MEP to measure the packet loss on the paths to and from a peer MEP. TWLM 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 TWLM message exchange.

```
                 TXp       RXp             TXc       RXc
  Sender     ----------------------------------------------
                  \       /\                \       /\
                   \      /      . . .       \      /
                 SLM \    / SLR            SLM \    / SLR
                     \/  /                     \/  /
  Reflector  ----------------------------------------------
                       TRXp                      TRXc
```

Figure 3 Two-Way Loss Measurement

The TWLM 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 rate by comparing the local counter values to the counter values received in the SLR messages.

The subscript ’c’ is short for current, and ’p’ is short for previous.

4.2.1. SLM Message Transmission

TWLM can be applied either proactively or on-demand.

A MEP that supports TWLM MUST support unicast transmission of SLM messages.

A MEP that supports TWLM 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. Therefore, the first packet of a probe instance is sent with the counter value set to 1.

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 TWLM 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 rate 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).

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

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

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.

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.

A sender MAY choose to monitor only the far-end packet loss, i.e., 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 DM protocol has two flavors, One-Way Delay Measurement (OWDM), and Two-Way Delay Measurement (TWDM).

5.1. One-Way Delay Measurement (OWDM)

OWDM is used for computing the one-way packet delay from one MEP to another. The packet format used in OWDM is referred to as 1DM, and is specified in Section 6.3.2. The OWDM message exchange is illustrated in Figure 4.

```
                          T1
      Sender    -------------------         ----> time
                       \
                        \ 1DM
                         \
                         \/
      Receiver  -------------------
                          T2
```

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:

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

Equation (4) implies that T2 and T1 are measured with respect to a common reference time. Hence, two MEPs running an OWDM 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 OWDM MUST support unicast transmission of 1DM messages.

A MEP that supports OWDM 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, T2. 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 OWDM MUST support reception of both unicast and multicast 1DM messages.

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

When OWDM 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 (TWDM)

TWDM 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 TWDM message exchange is illustrated in Figure 5.

```
                                T1          T4
                  Sender     ------------------------       ----> time
                                  \           /\
                                   \          /
                                DMM \        / DMR
                                     \/     /
                  Reflector  ------------------------
                                     T2    T3
```

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 TWDM MUST support unicast transmission of DMM messages.

A MEP that supports TWDM 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 TWDM 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:

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

While OWDM requires the two MEPs to be synchronized, TWDM allows the sender to calculate the two-way delay without being synchronized to the reflector.

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

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

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

When TWDM 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, i.e., 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 encapsulation is defined in [OAM-FRAMEWK], and is quoted in this document for clarity. For further details see [OAM-FRAMEWK].

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
.      Link Header              . Variable
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               | 8 bytes
+      TRILL Header             + fixed part of TRILL Header
|                               | _
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   |  \
|   DA   /   SA             |   |   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   |   |
|   Data Label              |   |    \  Flow Entropy
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    /  Fixed Size
.                               .   |
.                               .   |
|                               | _/
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       OAM EtherType           | 2 bytes
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
.   OAM Message Channel         . Variable
.                               .
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
.    Link Trailer               . Variable
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6 TRILL OAM Encapsulation

The OAM Message Channel used in this document is defined in [TRILL-FM], and has the following structure:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|MD-L | Version | OpCode        |  Flags        |TLVOffset      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.         OpCode-specific fields                                .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.         TLVs                                                  .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

LM 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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|MD-L | Ver (0) | OpCode        |  Flags (0)    |TLVOffset (16) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Sender MEP ID          |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Test ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Counter TX                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.         TLVs                                                  .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|MD-L | Ver (0) | OpCode        |  Flags (0)    |TLVOffset (16) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Sender MEP ID          | Reserved for Reflector MEP ID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Test ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Counter TX                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Reserved for SLR: Counter TRX (0)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.         TLVs                                                  .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|MD-L | Ver (0) | OpCode        |  Flags (0)    |TLVOffset (16) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Sender MEP ID          |       Reflector MEP ID        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Test ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Counter TX                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Counter TRX                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.         TLVs                                                  .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 DM 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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|MD-L | Ver (1) | OpCode        |  Reserved   |T|TLVOffset (16) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Timestamp T1                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved for 1DM receiving equipment (0)            |
|                      (for Timestamp T2)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.         TLVs                                                  .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|MD-L | Ver (1) | OpCode        |  Reserved   |T|TLVOffset (32) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Timestamp T1                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved for DMM receiving equipment (0)            |
|                      (for Timestamp T2)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Reserved for DMR (0)                      |
|                      (for Timestamp T3)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Reserved for DMR receiving equipment               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.         TLVs                                                  .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|MD-L | Ver (1) | OpCode        |  Reserved   |T|TLVOffset (32) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Timestamp T1                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Timestamp T2                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Timestamp T3                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Reserved for DMR receiving equipment               |
|                      (for Timestamp T4)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.         TLVs                                                  .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

7. Security Considerations

The security considerations of TRILL OAM are discussed in [OAM-REQ] and in [OAM-FRAMEWK]. General TRILL security considerations are discussed in [RFCTRILL]. This document does not inflict further security considerations.

8. 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 (LM) session or Delay Measurement (DM) session.

The LM session can be used to determine the performance metrics FLR, availability, and resiliency. The DM session can be used to determine the performance metrics FD, IFDV, FDR, and MFD.

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 PDU transmission frequency (the time between PDU transmissions).

- o The Measurement Interval is the time period over which measurements are gathered and then summarized. The Measurement Interval can align with the PM Session duration, but it doesn’t need to. PDUs during a PM Session are only transmitted during a Measurement Interval.

- o The Repetition Time is the time between start times of the Measurement Intervals.

```
          Measurement Interval     Measurement Interval
          (Completed, Historic)    (In Process, Current)
      |                        |
      |                        |
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      ^                 ^ ^                                         ^
      |                 | |                                         |
   Service Enabled     Message                                   Service
                        Period                                   Disabled
```

Figure 14 Relationship Between Different Timing Parameters

8.1. LM Statistics

- o Start Time : The time that the current Measurement Interval Started.

- o Elapsed Time : The time that the current Measurement Interval has been running, in 0.01 seconds.

- o Availability Statistics suspect : Whether the Measurement Interval has been marked as suspect. It’s started as FALSE at the start of a measurement. It’s set to true when there’s a discontinuity in the performance measurement during the Measurement Interval.

- o Availability Forward High Loss Interval: Number of high Loss intervals over this time in the forward direction.

- o Availability Backward High Loss Interval: The number of high loss intervals over this time in the backward direction.

- o Availability Forward Consecutive High Loss: The number of consecutive high loss intervals over the time in the forward direction.

- o Availability Backward Consecutive High Loss: The number of consecutive high loss intervals over the time in the backward direction.

- o Availability Statistics Forward Available: The number of availability indicators evaluated as available in the forward direction by this MEP during this Measurement Interval.

- o Availability Statistics Backward Available: The number of availability indicators evaluated as available in the backward direction by this MEP during this Measurement Interval.

- o Availability Statistics Forward Unavailable: The number of availability indicators evaluated as unavailable in the forward direction by this MEP during this Measurement Interval.

- o Availability Statistics Backward Unavailable: The number of availability indicators evaluated as unavailable in the backward direction by this MEP during this Measurement Interval.

- o Availability Statistics Forward Minimum FLR: The minimum one-way availability FLR in the forward direction.

- o Availability Statistics Backward Minimum FLR: The minimum one-way availability FLR in the backward direction.

- o Availability Statistics Forward Maximum FLR: The maximum one-way availability FLR in the forward direction.

- o Availability Statistics Backward Maximum FLR: The maximum one-way availability FLR in the backward direction.

- o Availability Statistics Forward Average FLR: The Average one-way availability FLR in the forward direction.

- o Availability Statistics Backward Average FLR: The Average one-way availability FLR in the backward direction.

8.2. DM Statistics

- o Start Time : The time that the current Measurement Interval Started.

- o Elapsed Time : The time that the current Measurement Interval has been running, in 0.01 seconds.

- o Availability Statistics suspect : Whether the Measurement Interval has been marked as suspect. It’s started as FALSE at the start of a measurement. It’s set to true when there’s a discontinuity in the performance measurement during the Measurement Interval.

- o Frame Delay Two Way : The two-way frame delay calculated by this MEP from the last received PM PDU.

- o Frame Delay Forward : The frame delay in the forward direction calculated by this MEP from the last received PM PDU.

- o Frame Delay Backward : The frame delay in the backward direction calculated by this MEP from the last received PM PDU.

- o Inter Frame Delay Variation Two Way : The two-way Inter frame delay variation calculated by this MEP.

- o Inter Frame Delay Variation Forward : The last one-way Inter frame delay variation in forward direction calculated by this MEP.

- o Inter Frame Delay Variation Backward : The last one-way Inter frame delay variation in backward direction calculated by this MEP.

- o Frame Delay Two Way Minimum : The minimum two-way frame delay calculated by this MEP for this Measurement Interval.

- o Frame Delay Two Way Maximum : The maximum two-way frame delay calculated by this MEP for this Measurement Interval.

- o Frame Delay Two Way Average : The average two-way frame delay calculated by this MEP for this Measurement Interval.

- o Frame Delay Forward Minimum : The minimum one-way frame delay in the forward direction calculated by this MEP for this Measurement Interval.

- o Frame Delay Forward Average : The average one-way frame delay in the forward direction calculated by this MEP for this Measurement Interval.

- o Frame Delay Forward Maximum : The maximum one-way frame delay in the forward direction calculated by this MEP for this Measurement Interval.

- o Frame Delay Backward Minimum : The minimum one-way frame delay in the backward direction calculated by this MEP for this Measurement Interval.

- o Frame Delay Backward Average : The average one-way frame delay in the backward direction calculated by this MEP for this Measurement Interval.

- o Frame Delay Backward Maximum : The maximum one-way frame delay in the backward direction calculated by this MEP for this Measurement Interval.

- o Inter Frame Delay Two Way Minimum : The minimum two-way inter-frame delay interval calculated by this MEP for this Measurement Interval.

- o Inter Frame Delay Two Way Maximum : The maximum two-way inter-frame delay interval calculated by this MEP for this Measurement Interval.

- o Inter Frame Delay Two Way Average : The average two-way inter-frame delay interval calculated by this MEP for this Measurement Interval.

- o Inter Frame Delay Forward Minimum : The minimum one-way inter-frame delay interval in the forward direction calculated by this MEP for this Measurement Interval.

- o Inter Frame Delay Forward Average : The average one-way inter-frame delay interval in the forward direction calculated by this MEP for this Measurement Interval.

- o Inter Frame Delay Forward Maximum : The maximum one-way inter-frame delay interval in the forward direction calculated by this MEP for this Measurement Interval.

- o Inter Frame Delay Backward Minimum : The minimum one-way inter-frame delay interval in the backward direction calculated by this MEP for this Measurement Interval.

- o Inter Frame Delay Backward Average : The average one-way inter-frame delay interval in the backward direction calculated by this MEP for this Measurement Interval.

- o Inter Frame Delay Backward Maximum : The maximum one-way inter-frame delay interval in the backward direction calculated by this MEP for this Measurement Interval.

- o Number of PDUs Sent : The count of the number of DMM PDUs sent.

- o Number of PDUs Received : The count of number of DMR received.

9. IANA Considerations

9.1. OpCode Values

IANA is requested to assign TRILL OAM OpCode values to the packet types defined in this document. The suggested OpCode values are identical to the ones defined in [Y.1731]:

45 : 1DM

46 : DMR

47 : DMM

53 : 1SL

54 : SLR

55 : SLM

10. Acknowledgments

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

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), May 2013.

[TRILL-FM] Senevirathne, T., Finn, N., Salam, S., Kumar, D., Eastlake, D., Aldrin, S., Li, Y., "TRILL Fault Management", draft-tissa-trill-oam-fm (work in progress), May 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.

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 USA R&D
155 Beaver Street
Milford, MA 01757 USA
Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

TRILL Working Group
INTERNET-DRAFT
Intended Status: Informational

Expires: January 13, 2014

Yizhou Li
Donald Eastlake
Weiguo Hao
Huawei Technologies

July 12, 2013

Problems of Active-Active connection at the TRILL Edge
draft-yizhou-trill-active-active-connection-prob-00

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 and link technology between TRILL switches. Active-active at the TRILL edge is the extension, in so far as practical, of these characteristics to end stations that are multiply connected to a TRILL campus. This informational document discusses some of the high level problems to be overcome in providing active-active at the TRILL edge.

Status of this Memo

Copyright and License Notice

Table of Contents

1 Introduction

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

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 inbetween. By carefully selecting different RBridge as appointed forwarder for different set of VLANs, load spreading over different edge RBidges across different VLANs can be achieved.

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

```
             ----------------------
            |                      |
            |   TRILL Campus       |
            |                      |
             ----------------------
                 |       |    |
            -----        |     --------
           |             |             |
        +------+      +------+      +------+
        |      |      |      |      |      |
        |(RB1) |      |(RB2) |      | (RBk)|
        +------+      +------+      +------+
          |..|          |..|          |..|
          |  +----+     |  |          |  |
          |   +---|-----|--|----------+  |
          | +-|---|-----+  +-----------+ |
MC-       | | |   +------------------+ | |
LAG1--->(| | |)                    (| | |) <---MC-LAG2
        +-------+    .  .  .       +-------+
        | CE1   |                  | CEn   |
        |       |                  |       |
        +-------+                  +-------+
```

Active-Active connection is useful when we want to achieve the following requirements though MC-LAG implementation varies by vendor.

- Flow rather than VLAN based load balancing is required.

- Rapid failure recovery. 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-appoint the forwarder for specific VLANs may be required. Such procedures takes time in the scale of seconds. Active-Active connection should minimize the frame loss and recovery time in failure.

3. Problems in active-active connection at the edge

This sections present the problems needed to be addressed in active-active connection scenario.

3.1 Frame duplications

When an MC-LAG is formed to multiple RBridges, there may be a potential duplication of the frame to be received by the a CE. Two possible scenarios are presented as follows.

1. Looping back: CE1 forwards a multi-destination frame from a user device. As shown in Figure 1, the frame enters the TRILL campus via a member of an MC-LAG (say RB1) and then is forwarded through the campus to another member (say RB2) of the same MC-LAG. Then CE1 receives a duplicated copy from RB2.

2. Duplication from remote: A remote RBridge sends a multi-destination frame of VLAN x. All members of MC-LAG1 will receive the frame. As each of them thinks it is the appointed forwarder for all VLANs, they would all forward the frame to CE1. The consequence is CE receives multiple copies.

Frame duplication only happens in multi-destination frame forwarding. Unicast does not have this issue.

3.2 Address flip-flop

Consider RB1 and RB2 using their own nickname as source nickname to ingress data frame into a TRILL campus. As shown by Figure 1, CE1 may send a data frame with the same source MAC address to any member RB of MC-LAG1. If the egress RBridge receives TRILL packet from different ingress RBridge RBridges but with same same source MAC

address, it learns different address correspondence from the 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. Some 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 Packet drop due to RPF check

In order to solve the problems above, a pseudonode nickname [TRILLPN] solution was proposed. The basic idea is to represent all member links of the MC-LAG as a virtual RBridge with single pseudonode nickname. Any member RBridge of the MC-LAG should use this pseudonode nickname rather than its own nickname as ingress nickname when inject TRILL data frames. It solves the abovementioned problems pretty well; however, it introduces another issue: packet drop due to RPF check.

When forwarding multi-destination frame, different member RBridges of an MC-LAG may choose the same tree. A random RBridge RBn in TRILL campus may receive the frame on single tree from the pseudonode nickname on different incoming ports. RPF check fails in this case. Frames will be dropped.

3.4 Loops

Active-Active connection does not introduce extra looping risk as MC-LAG is just like a single link. So a frame will not keep geting ingress and egressed to/from the TRILL campus via a single MC-LAG link in normal situation. However we do need to pay attention that any solutions for active-active connection scenario make sure the campus is loop-free.

3.5 Member RBridges info synchronization

When multiple edge RBridges are bundled as an MC-LAG to make CE multi-homed to TRILL campus, it is necessary to make sure the RBridges are aware of the status of each link in MC-LAG. Synchronization of information is necessary.

1. Member RBridges configuration synchronization: it is unavoidable to synchronize the configuration parameters among edge RBridges of an MC-LAG. Such configuration may include system ID, system priority, port key, port priority, partner information, etc. If abovementioned

[TRILLPN] and/or [CMT] was employed, there are more configurations to be synchronized, for instance, pseudonode nickname of the virtual RBridge. Without synchronization mechanism, we have to manually provision each member RBridge to guarantee consistency. In addition, some of the configuration may dynamically change during failure, for instance, tree-id selected by member RBridges [CMT]. Manual inconsistency check is not applicable in this case.

2. Member RBridges state synchronization: link failure or node failure on a member RBridge may introduce packet loss. Link failure includes both access port and trunk port link failure. When failure occurs, MC-LAG may need to invoke re-selection logic to spread the traffic across the rest links/nodes. Therefore fast detection and failure recovery is required upon state synchronization. Some mechanism could be employed, for example, TRILL BFD support[TRILLBFD]. Trunk port and node failure can be detected it. However access port/link failure needs some special care. An RBridge that has an access port/link failure should notify the other members RBs with port information to make them adjust the corresponding MC-LAG.

3. Member RBridges learnt MAC address synchronization: it is required that member RBs share the MAC address and egress nickname correspondence they have learnt. By such synchronization, flooding due to unknown unicast can be reduced.

If some inter-chassis protocol is employed among member RBridges for MC-LAG member discovery, info synchronization and failure handling, we need to make sure it can run smoothly over TRILL campus. The protocol may use IP address to identify the other members. We need to make sure such packets can be correctly TRILL encapsulated.

If no such inter-chassis protocol is available, TRILL has to provide its own mechanisms to support the information synchronization.

4 Current Work

There have been some solution drafts presented in TRILL WG. [TRILLPN], [CMT] and [TRILLBFD] address parts of the problems above.

5 Security Considerations

This draft presents the problems in a particular scenario. It 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.

6 References

5.1 Normative References

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

[RFC6325] Perlman, R., et.al. "RBridge: 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] Eastlake, D. et.al., "RBridge: Appointed Forwarder", RFC 6439, November 2011.

5.2 Informative References

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

[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

[TRILLBFD] V. Manral., et al., "TRILL (Transparent Interconnetion of Lots of Links): Bidirectional Forwarding Detection (BFD) Support", draft-ietf-trill-rbridge-bfd-07.txt work in Progress, July 2012

[8021AX] IEEE, "Link Aggregration", 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: +86-25-56625375
EMail: liyizhou@huawei.com

Donald Eastlake
Huawei R&D USA
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