Abstract

This document presents a solution suite for TRILL data center sites to be connected over WAN networks. TRILL protocol is primarily designed to work within intra-data centers. Connecting different sites over WAN using overlay tunnel protocols is the primary method employed at present. Though this presents a simple mechanism to extend the LAN sites to be interconnected, it also brings in the problem of scalability for TRILL nicknames exchanged between sites, latency, duplication of traffic etc. This draft proposes a way to extend the TRILL sites without having to reveal the data of the LAN like customer MAC’s or provide MAC’s over the WAN, but to establish connections between various sites by extending routing protocol to exchange minimal information, thus reducing the information flow to the required sites only. Document also proposes BGP routing protocol extensions as an example to establish paths and information about the essential RBridges nicknames, over WAN networks like MPLS.

Status of this Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
1 Introduction
1.1 Terminology
2. Solution overview
2.1 Site inter-connection
2.2 Requirements overview
2.2 TRILL campus extension
2.3 TRILL nickname exhaustion
3. Solution comparison analysis
3.1 TRILL campus extension
3.2 TRILL campus interconnection with E-VPN and PBB-EVPN
3.3 TRILL campus interconnection over VPN’s
4. Operational Overview
4.1 Campus and Backbone Areas
4.2 Unicast forwarding
4.3 Multicast forwarding
4.4 MAC learning
4.5 TRILL nickname aggregation
4.6 Route advertisement with BGP
5. TRILL campus inter-connectivity
5.1 Route advertisement
5.2 Inter-site nickname exchange
5.3 Border RBridge capability exchange
1 Introduction

TRILL protocol is primarily designed as an intra-datacenter protocol by leveraging the routing functionality to interconnect bridges. Traditional Ethernet networks provided a single path for forwarding the traffic, which is usually derived using protocols like Spanning Tree. TRILL provided a way to utilize multiple links for forwarding, thus utilizing the resources effectively. Even though TRILL is a new protocol, it seamlessly integrates with legacy bridging networks without having to forklift upgrade of all the bridges to support TRILL. By not having to learn the MAC addresses of end stations by intermediate devices, provided a powerful way to interconnect bridges within a datacenter and maximizing the resource usage and providing multipath usage option.

TRILL enabled network creates efficiency by having reduced forwarding table size. By doing TRILL nickname based forwarding created a layer of abstraction and much easier to implement the protocol. This enabled to address the scalability of a L2 domain, where thousands of R Bridges could exist to meet the needs of a datacenter. By leveraging IS-IS protocol, the information exchange and leveraging the path computation technology brought forth a new paradigm into bridging technology. TRILL Base Protocol Specification [RFC6325] specifies a tree based paradigm to forward broadcast and multicast traffic as well as unknown unicast traffic.

Even though the TRILL is enabled within a datacenter and is not primarily designed to work over WAN, there is a need to interconnect various TRILL data sites. The same datacenter provider could be having multiple TRILL sites and these TRILL enabled datacenter sites could run independently or could share resources in order to cater to the needs of customers. As such, there exist few proposals based on overlay technologies which interconnect these sites but those solutions require MAC learning at the edge R Bridges and stripping of TRILL nickname on the frame. Another option is to interconnect these TRILL sites using Pseudowires and making a huge TRILL site. This is useful option but the downside of this is when provider would like to maintain independent sites and exchange only the required data to be shared across sites, it becomes complicated to maintain the networks.

This draft solves these core problems of interconnection, site independence, dynamic information exchange and setting up of connections over WAN, leveraging existing WAN technologies like MPLS, etc. Though the primary goal is effective interconnection, there are various efficient schemes, which could enable seamless TRILL network deployments, are also be presented. Solution covers both unicast and multicast data traffic.
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].

2. Solution overview

This section provides the high-level overview of the solution to the problems in various scenarios. More detailed representation of the solution is covered in the later sections of the draft.

TRILL site or TRILL campus uses IS-IS to setup the RBridge interconnection. A RBridge knows how to reach another RBridge within the campus. When two TRILL campuses are interconnected, one could visualize it in two different perspectives. First one is a merger of two TRILL campuses into one. This requires each RBridge to know about the other and the IS-IS should be able to compute the shortest paths from one to another. The downside of this model is that information exchange explosion and the size of IS-IS db and number of PDU’s being exchanged could increase exponentially.

The second perspective is to have these two TRILL campuses being interconnected over a WAN, but their functioning nature is independent to each other. The two campuses exchanges only the required information between border RBridges of the campuses. This will be more optimal and leads to interconnecting multiple campuses without having to redesign the whole network to ensure uniqueness and identity of RBridges.

Solution being proposed is an option of maintaining site independency with a simplified solution and modeled around the existing and proven technologies. Some of the enhancements were already proposed in other drafts like multilevel TRILL [TISSA-MLEVEL] and the solution leverages by extending those definitions as necessary. The solution addresses the following areas described in the following sections

2.1 Site inter-connection

Each TRILL campus is considered as an independent site or an L1 IS-IS domain. These TRILL campus sites are interconnected over WAN. Each area will have an appointed border RBridges. These RBridges exchange the information of other border RBridges of different TRILL campus sites to establish connection with each other. In order to exchange the information, the route has to be established. Extension of BGP protocol is done to exchange the border RBridges information and establish a route or vpn/mvpn connection between each of the border
RBridges. More details of the extension are detailed in the below section. These sites are interconnected either over IP or MPLS. As MPLS is a mature WAN technology, this draft references the solution based on MPLS. This does not preclude that other technologies could not be used.

2.2 Requirements overview

There are various requirements necessary to be met in order to provide a seamless TRILL inter-connectivity across campuses and datacenter. Some of the important requirements are as follows

- Extend TRILL technology over interconnect
- Ability to provide the same fast convergence for mobility as it does in intra-TRILL campus
- Ability to work with various WAN technologies
- Option for dynamic establishment of connectivity across sites
- Minimal changes to protocols and definitions
- Backward compatible to existing networks and their functions.

2.2 TRILL campus extension

By interconnecting TRILL campus sites over WAN, one could extend the L1 area, but that would cause other issues as detailed in the earlier section. With this solution, all of the TRILL nicknames of one campus are not exchanged with other campuses. Instead, there will be RBridges which get appointed, so only those specific nicknames are exchanged with others. Also one could create another hierarchical model like VPN, where participating appointed RBridges for that VPN could be exchanged with other campuses belonging to that VPN. Appointed or designated RBridges information will be exchanged with other campus site using the Affinity TLV extension. Detailed description on how to create and the usage are detailed in multilevel draft [TISSA-MLEVEL]. When each campus site creates this information and exchanged with other campus sites of the network or VPN, this could be used for two purposes, 1. A VPN specific WAN paths could be created between campus sites. 2. The data distribution could be optimized without having the need to be broadcast the data frames from one campus into every campus site.

2.3 TRILL nickname exhaustion
Though this draft is not meant to provide solution for TRILL nickname exhaustion, it enables provider to deal with the problem effectively and not having to re-design the network, every time a new campus is interconnected. The proposed solution has RBridges which are not required to be exposed outside of the campus and there are other RBridges which are also known as border RBridges. These border RBridges nicknames are unique globally. Rest of the RBridges nicknames are significant locally, that includes the appointed RBridges nicknames for a VPN. When a frame has to be forwarded to an RBridge which resides in another campus, the originating RBridge only knows how to get to the borderRBridge. This border RBridge should have the list of RBridges of other campus sites and thus could select the appropriate MPLS LSP or MVPN or PW and encapsulate the TRILL frame with the label header and forward over that. More details are covered in the unicast and multicast sections of the detailed solution.

3. Solution comparison analysis

As eluded to in the earlier sections, there are various methods on interconnecting different TRILL campus sites. Before going into the details of proposed solution a close examination of some of the proposed solutions, provides better perspective of this solution.

3.1 TRILL campus extension

In this model TRILL campuses are connected over WAN using technologies like PW. This is the most simplest way of interconnecting the sites. When campuses are interconnected, the TRILL campus will get expanded and each RBridge could reach each other. The main criteria for this will be to maintain unique nickname for RBridges.

```
+-----------------+     +-----------------+     +-----------------+
| TRILL Campus    |     |  WAN             |     | TRILL Campus    |
|-----------------|     |-----------------|     |-----------------|
| RB1             |     | BRB1===         |     | ===BRB2        |
|                 |     |                 |     |                 |
|                 |     |                 |     |                 |
+-----------------+     +-----------------+     +-----------------+
```

As shown in the figure above, two TRILL campuses are interconnected over WAN. Border RBridges establish connection over WAN using PW or other WAN technologies. All the nicknames within each campus sites have to be unique. The WAN in this case is transparent to the TRILL
campuses and the path computation doesn’t involve WAN component, instead it will be like one TRILL campus. When RB1 originates a TRILL frame destined to RB2, it traverses over BRB1 and BRB2 and reaches RB2.

This solution is workable when the campuses are small and do NOT need to change or requires interconnecting more TRILL campuses. The other downside for this model is, when two campuses are interconnected and there is overlap of nicknames, the network has to be re-designed to eliminate the duplicate nicknames and make each RBridge to have a unique nickname.

3.2 TRILL campus interconnection with E-VPN and PBB-EVPN

TRILL campuses could be extended over WAN using E-VPN and PBB-EVPN.

The [PBB-EVPN] draft proposes interconnection details on how two TRILL campuses could be interconnected using the E-VPN technology. In this a new BGP route is advertised for reachability of TRILL RBridges. This technique leverages the PBB technology and also enables to retain TRILL header but is recommended to avoid transmitting TRILL encapsulated frames over the WAN links. The primary downside of this method is the requirement for edge RBridges to learn MAC Addresses in order to resolve the adjacency.

3.3 TRILL campus interconnection over VPN’s

In this method, TRILL campus sites could be interconnected over VPN’s.
These VPN’s could be established statically or dynamically. In order to establish dynamically, the border RBridges needs to exchange information of the nicknames and connect different sites. The hierarchical model like H-VPLS could be established as well. [PBB-EVPN] draft provides some details on how to achieve this, but still requires border RBridges to exchange MAC information and resolution for L2 adjacency. One other option is much similar to the first model where campuses exchange TRILL nicknames between campuses over VPN’s. Though this model groups different sites according to the way VPN’s are configured, avoiding flooding of TRILL nicknames or site independency cannot be achieved.

4. Operational Overview

4.1 Campus and Backbone Areas

Each TRILL campus will be assigned a border RBridge. This is identified using the ‘Attached’ bit in the IS-IS PDU. The border RBridge has list of the RBridges of each campus site. These list of bridges are exchanged using the TRILL nickname aggregation sub-TLV. Details of the sub-TLV are detailed in the below section.

Every TRILL campus campus need not exchange all the RBridge nicknames with other campuses. Let us take the scenario of campus A to be interconnected with campus B. In campus A, there are RBridges
RB1...RB10, which are interconnected in L1. These nicknames are not tunneled or exchanged with other L1 campus sites. Similarly campus B has RB11...RB20 and need not be distinct from campus A RBridge nicknames. So, if a new campus is connected to the domain, there is no need for the network to redesigned or restructured.

The RBridges at each campus advertise the information to other campus to establish a route/MPLS LSP between campus sites. A new BGP extension as defined below is sent out. This reachability TLV is received by other border RBridges and establish the MPLS LSP between them.

The border RBridges will have the complete information of its campus RBridges. Not all of the RBridges nicknames need not be advertised globally. So, the globally exchanged nicknames of RBridges should be unique across campuses. Depending on the policy established, these Border RBridges will exchange the TRILL information with other campus border Bridges, over the routes/MPLS LSP established between different campuses. In IS-IS domain the equivalent of this is the L2 backbone area, which in this case, is established over WAN.

4.2 Unicast forwarding

If the destination TRILL nickname is not known, the originating or transit RBridges forwards it to border RBridge. As the border RBridge has all the nicknames of each campus, it forwards the frame to the right campus border RBridge, which in turn could forward within its campus as per base protocol specification [RFC6325]. In the case where the destination is unknown, the frame is flooded to each campus. Using the MAC learning procedures, the associated RBridge will be learnt for the subsequent frames to be forwarded as unicast, instead of flooding. Flooding into various campuses or TRILL data sites happen only if the the frame is of global ID based on VLAN identification.

4.3 Multicast Forwarding

Whether the traffic scope is local or global across campuses is identified by VLAN or port or fine-grain label. If the traffic is to be forwarded within campus, it is done using the local tree. But if the forwarding has to be done globally, it needs to use the global tree. When the frame has global context, it will be flooded into other TRILL sites as well.

In MPLS the multicast networks are established within the core networks. In the case of RBridges, which are part of the customer networks and do not participate in the service provider networks and their topologies, the multicast tree could be built using IP
multicast or leverage MVPN services offered by the service provider. The global trees are established between border RBridges with the help of information exchanges between border RBridges. As the IS-IS is limited to individual campus sites, the information for the backbone tree over WAN has to be exchanged between border RBridges either as an IP multicast PIM message or specific TLV indented for other campus border RBridges. More details on this will be added in the later versions of the draft.

In the above figure when the multicast frame has to be sent from campus 1 to campus 2 and 3, the frame arrives at border RBridge BRB1. With the default global tree between border RBridges of different campuses, the forwarding is setup to egress the frame or replicated over MPLS LSP’s to all other campus sites. If the frame is destined for non-default global tree, the appropriate MPLS LSP(s) are identified and the frame is forwarded accordingly.

Once the frame is reached on the border router of the campuses, the frame is locally multicast forwarded. The same technique as employed in the multilevel draft [TISSA-MLEVEL] is used here as well.

If mVPN services are deployed interconnecting campus sites, the multicast tree is built over these services based on the customer VLANs.
4.4 MAC learning

When a frame is to be forwarded from customer MAC A to customer MAC B, the frame is set as unknown unicast frame over TRILL networks. If the MAC A and MAC B are connected over WAN, the frame is transmitted over WAN to the other campus. When the frame is reached at the RBridge connecting to MAC B, it will learn about the originator RBridge for MAC A. While responding, the egress RBridge know the originating RBridge, it will unicast the frame to the originator.

4.5 TRILL nickname aggregation

Nicknames are allocated or assigned to RBridges in a given campuses using various methods. It could be OSS, CLI or could be a dynamic control protocol which configures the nicknames. As the nicknames are confined to each L1 area, the nickname management, when sites are connected over WAN, it is essential to optimize the name allocation in order to use the name space effectively. Name allocation is not in the scope of this draft. If there is a necessity, the topic could be considered in the later revisions of the draft. For this version we do recommend some of the optimization techniques for nickname allocation defined in the multilevel draft [TISSA-MLEVEL].

Each border RBridges needs to exchange the nicknames of each campuses with other border RBridges. As the border RBridges are connected over various types of WAN networks, mandating enhancement to a specific protocol is deemed not the right approach. As the information exchange has to be done, certain characteristics for the data exchange have to be met.

- The amount of data exchange has to be minimal and optimized.
- The information exchange has to be quick.
- Conflicts and duplicate information flow has to be avoided

This draft proposes a generic TLV, which is to be exchanged between border RBridges. If the nickname allocation is done in terms of ranges, the same could be exchanged between border RBridges, seamlessly. As the TLV has to be terminated at the border RBridges, it is better suited to be sent as a unicast message to the neighboring border RBridge. This could be sent as an IP message with TRILL header containing the target border RBridge. More details on how to encapsulate and process the frame should be in the later versions of the draft.

4.6 Route advertisement with BGP
BGP route advertisement is to connect border RBridges. As described in earlier sections, each campus site exchanges the TRILL nicknames with other campuses. These nicknames are not leaked into the campus but will be maintained at the border RBridges. The connectivity between these border RBridges is similar to L2 backbone of IS-IS but in this case it is over WAN and IS-IS is absent.

This draft, unlike some other drafts, do not propose to de-capsulate the TRILL frame to learn the MAC addresses. Instead, it proposes to use nicknames themselves to be exchanged between sites.

A new BGP route advertisement is defined to advertise the border RBridge nickname with the other border RBridges in different campuses. This advertisement will let other campuses know its MPLS label, its nickname, IP address etc. Once the route is established, further information of campus nicknames etc could be exchanged to establish the inter-connectivity of the TRILL campuses.

5. TRILL campus inter-connectivity

The primary reason to interconnect TRILL campuses is to maintain geographically distant, segmented sites and customer specific segregation possible by interconnecting and not having to redo the network and campus redesign for every change and need. With customers being mobile or services offered by service providers could be re-located depending on the time-zones and resource availability, restricting to a specific site is a thing of the past.

These constraints brought forth the need to have different sites interconnected over the WAN, be it MPLS or VPLS or IP and to provide the services on demand to meet the needs of customers and their data center needs. As TRILL has proven to be an effective protocol by bringing routing technologies into bridges or L2 forwarding, the short coming of TRILL interconnect is the immediate need. As eluded to in the earlier sections on different kinds of solutions, meeting all the needs of the TRILL DCI as laid out in the requirements section, is the primary goal of this draft.

5.1 Route advertisement

Border RBridges only participate in interconnecting various TRILL campuses. These border RBridges are elected or identified as described in the earlier section i.e. using IS-IS protocol advertisement. These border RBridges, when required to interconnect with other campuses, advertise the route to other site border RBridges using the BGP enhancement. Upon receipt of the route advertisement, the MPLS LSP or IP path or Pseudowire is established between RBridges and they could communicate with each other and
exchange other information.

If L2 connectivity is to be used with protocols like VPLS, a similar method could be employed, where the PWE3 could be established between border RBridges. More details to be added in the later versions of the draft.

5.2 Inter-site nickname exchange

There are three types of nicknames which are exchanged between border RBridges.

- Nickname of border RBridges
- Nicknames of RBridges for each campus
- Nicknames of RBridges which are part of a specific customer VLAN or VPN

The nickname aggregation TLV is used as payload to be exchanged between border RBridges. This information is used to establish interconnectivity between TRILL sites per customer VLAN or default global tree.

5.3 Border RBridge capability exchange

An additional capability TLV is defined to exchange info on what each of the border RBridge is capable of. This is very essential for forward and backward capability. Capability information not only indicates the capability version but could also force the interconnection to be restricted as per the policy set by the customer. Some of the capability advertisements are as follows.

- Version.
- default nickname resolution
- connect more campuses
- active-active link support
- Ability to support multicast forwarding

5.4 TRILL adjacency resolution

When a frame is to be forwarded from one campus to another, the adjacency resolution has to be done on the border RBridge. When TRILL nicknames are advertised from one border RBridge to another, the
border RBridge keeps the database of all the nicknames. The MPLS LSP’s between each RBridges are established by the route advertisements. VPN specific services could be established based on the customer VLAN ID’s and also the exchange of nicknames per VPN between border RBridges.

Once the frame is received on the border RBridge, it will look in the forwarding table to identify the next hop. The adjacency information could indicate an MPLS LSP with a specific label encapsulation. For VPN, there will be additional VPN label in the label stack. The TRILL frame is encapsulated, without removing the TRILL header and is forwarded over the MPLS LSP.

5.5 Forwarding of data frames

The TRILL frames are forwarded as per the base protocol [RFC6325] within a campus site. The forwarding of the frames from TRILL campus to campus will be over MPLS LSP’s or IP or whichever WAN connection is established between border RBridges. The encapsulation of the Frame with WAN header is based on the adjacency resolution made in the forwarding on the border RBridge.

6. Proposed additions and extensions

There are certain extensions being proposed in this draft to interconnect TRILL campuses or datacenters. These include new additions to routing and also new TLV’s to exchange information between border RBridges. There are few references to the extensions proposed in other drafts which are used in this draft as well.

6.1 BGP extension

A new BGP route advertisement is done to exchange and establish route/MPLS lsp between border RBridges.

```
+---------------------------------------+
| RD (8 octets)                      |
+---------------------------------------+
| Originating RBridge MAC address     |
+---------------------------------------+
| Originating RBridge IP address(v4/v6)|
+---------------------------------------+
| Nickname Length (1 octet)           |
+---------------------------------------+
| RBridge Nickname (2 octets)         |
+---------------------------------------+
| MPLS Label (n * 3 octets)           |
+---------------------------------------+
```
6.2 Border RBridge capability TLV

This TLV as defined in earlier section, defines the capability of a border RBridge, to be exchanged with other border RBridges for seamless inter-working across campus sites.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type = <TBD>             |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Version                   |  Flags                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Flags                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Flags                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Definition of flag bits will be identified and defined later.

6.3 TRILL nickname aggregation sub-TLV

The nickname aggregation TLV defined in multilevel draft [TISSA-MLEVEL] is used in advertising the nicknames into other border Routers. Some new additions or changes will be proposed in later versions of the draft.
7 Security Considerations

<Security considerations text>

8 IANA Considerations

<IANA considerations text>

9 References

9.1 Normative References


9.2 Informative References


Authors’ Addresses

Aldrin, et.al. Expires September 3, 2012
Sam Aldrin  
Huawei Technologies  
2330 Central Express Way  
Santa Clara, CA 95951  
Email: aldrin.ietf@gmail.com

Tissa Senevirathne  
CISCO Systems  
375 East Tasman Drive  
San Jose CA 95134  
Phone: 408-853-2291  
Email: tsenevir@cisco.com

Ayan Banerjee  
CISCO Systems  
425 East Tasman Drive  
San Jose CA 95134  
Phone: 408-527-0539  
Email: ayabaner@cisco.com

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

Santiago Alvarez  
CISCO systems  
Email: saalvare@cisco.com
Directory Assisted RBridge Edge
draft-dunbar-trill-directory-assisted-edge-05.txt

Status of this Memo

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

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

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

This Internet-Draft will expire on September 11, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Abstract
RBridge edge nodes currently learn the mapping between MAC addresses and their corresponding RBridge edge nodes by observing the data packets traversed through. When ingress RBridge receives a data packet with its destination address (MAC&VLAN) unknown, the data packet is flooded across RBridge domain. When there are more than one RBridge ports connected to one bridged LAN, only one of them can be designated as AF port for forwarding/receiving traffic for each LAN, the rest have to be blocked for that LAN.

This draft describes the framework of using directory assisted RBridge edge to improve TRILL network scalability in data center environment.

Conventions used in this document

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

Table of Contents

1. Introduction ................................................ 3
2. Terminology ................................................. 3
3. Impact on RBridge domain of massive number of hosts in Data Center ............................................. 4
   3.1. Issues of Flooding Based Learning in Data Centers ........ 4
   3.2. Some Examples .......................................... 5
4. Benefits of Directory Assisted RBridge Edge in DC Environment .............................................. 7
5. Generic operation of Directory Assistance.................... 8
   5.1. Information in Directory Servers for TRILL............... 8
   5.2. Push Model ............................................. 8
   5.3. Pull model: ........................................... 10
6. Conclusion and Recommendation................................ 11
7. Manageability Considerations................................ 11
8. Security Considerations.................................... 11
9. IANA Considerations ...................................... 11
10. Acknowledgments .......................................... 11
11. References ............................................... 12
Authors’ Addresses ........................................... 12
Intellectual Property Statement................................. 13
Disclaimer of Validity ........................................ 13
1. Introduction

Data center networks are different from campus networks in several ways, in particular:

1. Data centers, especially Internet or multi-tenant data centers, tend to have large number of hosts with a wide variety of applications.
2. Topology is based on racks and rows. Hosts assignment to Servers, Racks, and Rows is orchestrated by Server/VM Management system, not at random.
3. Rapid workload shifting in data centers can accelerate the frequency of one physical server being re-loaded with different applications. Sometimes, applications re-loaded to one physical server at different time can belong to different subnets.
4. With server virtualization, there is an ever-increasing trend to dynamically create or delete VMs when demand for resource changes, to move VMs from overloaded servers, or to aggregate VMs onto fewer servers when demand is light.

Both 3) and 4) above can lead to hosts in one subnet being placed under different locations (racks or rows) or one rack having hosts belonging to different subnets.

This draft describes why and how Data Center TRILL networks can be optimized by utilizing a directory assisted approach.

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

STP: Spanning Tree Protocol

RSTP: Rapid Spanning Tree Protocol

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

VM: Virtual Machines

3. Impact on RBridge domain of massive number of hosts in Data Center

3.1. Issues of Flooding Based Learning in Data Centers

It is common for Data Center networks to have multiple tiers of switches, e.g. one or two Access Switches for each server rack (ToR), aggregation switches for some rows (or EoR switches), and some core switches to interconnect the aggregation switches. Many aggregation switches deployed in data centers are high port density switches. It is not uncommon to see aggregation switches interconnecting hundreds of ToR switches.

![Diagram of typical Data Center network design]

Figure 1: Typical Data Center Network Design
When TRILL is deployed in a data center with large number of hosts, with the possibility of hosts in one subnet/VLAN being placed under multiple edge RBridges and each edge RBridge having hosts from different subnets/VLANs, the following problems will occur:

Unnecessary filling of slots in MAC table of edge RBridges, due to edge RBridge receiving broadcast/multicast traffic (e.g. ARP/ND, cluster multicast, etc.) from hosts under other edge RBridges that are not actually communicating with any hosts attached to the RBridge.

Some edge RBridge ports being blocked for user traffic when there are more than one RBridge ports connected to one bridged LAN. When there are multiple RBridge ports connected to a bridged LAN, only one, i.e. the AF port, can forward/receive traffic for that bridged LAN (i.e. VLAN), the rest have to be blocked for forwarding/receiving traffic for that VLAN. When a rack has dual uplinks to two different ToR switches (RBridge Edges), which is very common in data center environment, some links can’t be fully utilized.

Packets being flooded across RBridge domain when their DAs are not in ingress RBridge’s cache.

In an environment where VMs migrates, there is higher chance of cached entries becoming invalid, causing traffic to be black holed or re-flooded by the egress RBridge. If VMs send out gratuitous ARP/ND or IEEE802.1Qbg’s VDP upon arriving at new locations, the ingress nodes might not have the MAC entries for the newly arrived VMs, causing more unknown flooding.

3.2. Some Examples

Consider a data center with 1600 server racks. Each server rack has at least one ToR switch. The ToR switches are further divided to 8 groups, with each group being connected by a set of aggregation switches. There could be 4 to 8 aggregation switches in each set to achieve load sharing for traffic to/from server racks. If TRILL is to be deployed in this data center environment, let’s consider following two scenarios for the TRILL domain boundary:

Scenario #1: TRILL domain boundary starts at ToR switches:
If each server rack has one uplink to one ToR, there are 1600 edge RBridges. If each rack has dual uplinks to two ToR switches, then there will be 3200 edge RBridges.

In this scenario, the RBridge domain will have more than 1600 (or 3200) + 8*4 (or 8*8) nodes, which is quite a large IS-IS domain. Even though a mesh IS-IS domain can scale up to thousands of nodes, it is very challenging for aggregation switches to handle IS-IS link state advertisement among hundreds of parallel ports.

Scenario #2: TRILL domain boundary starts at the aggregation switches:

With the same assumption as before, the number of nodes in RBridge domain will be less than 100, and aggregation switches don’t have to handle IS-IS link state advisements among hundreds of ports.

But in this scenario, aggregation switches’ downstream ports/links to ToR switches form the bridged LAN with links from ToR switches to servers. With aggregation switches being the RBridge edge nodes, multiple RBridge edge ports could be connected to one bridged LAN. To avoid potential loops TRILL requires only one of multiple RBridge edge ports connected to one VLAN being designated as Appointed Forwarder (AF port) for forwarding native traffic across RBridge domain for that VLAN. That means other ports/links are blocked for native frames in that VLAN.

There is also possibility of loops on the bridged LAN attached to RBridge edge ports unless STP/RSTP is running. Running traditional Layer 2 STP/RSTP on the bridged LAN in this environment may be overkill because the topology among the ToR switches and aggregation switches is very simple.

In addition, the number of MAC&VLAN<->RBridgeEdge Mapping entries to be learned and managed by RBridge edge node can be very large. In the example above, each edge RBridge has 200 edge ports facing the ToR switches. If each ToR has 40 downstream ports facing servers and each server has 10 VMs, there could be 200*40*10 = 80000 hosts attached. If all those hosts belong to 1600 VLANs (i.e. 50 per VLAN) and each VLAN has 200 hosts, then under the worst case scenario, the total number
of MAC&VLAN entries to be learned by the RBridge edge can be 1600*200=320000, which is very large.

4. Benefits of Directory Assisted RBridge Edge in DC Environment

In data center environment, applications placement to servers, racks, and rows is orchestrated by Server (or VM) Management System(s). I.e. there is a database or multiple ones (distributed model) which have the knowledge of where each host is located. If that host location information can be fed to RBridge edge nodes, in some form of Directory Service, then RBridge edge nodes won’t need to flood data frames with unknown DA across RBridge domain.

Avoiding unknown DA flooding to RBridge domain is especially valuable in data center environment because there is higher chance of an RBridge edge receiving packets with unknown DA and broadcast/multicast messages due to VM migration and servers being loaded with different applications. When a VM is moved to a new location or a server is loaded with a new application with different IP/MAC addresses, it is more likely that the DA of data packets sent out from those hosts are unknown to their attached RBridge edges. In addition, gratuitous ARP (IPv4) or Unsolicited Neighbor Advertisement (IPv6) sent out from those newly migrated or activated hosts have to be flooded to other RBridge edges which have hosts in the same subnets.

The benefits of using directory assistance include:

- Avoid flooding unknown DA across RBridge domain. The Directory enforced MAC&VLAN <-> RBridgeEdge mapping table can determine if a data packet needs to be forwarded across RBridge domain.

- When multiple RBridge edge ports are connected via bridged LAN to hosts (servers/VMs), a directory assisted RBridge edge won’t need to flood unknown DA data frames to all ports of the RBridge edge. Under this circumstance, there is no chance for those data frames looping among multiple ports of RBridge edge. Therefore, it is no longer necessary to designate one Appointed Forwarder among all the RBridge Edge ports connected to a bridge LAN, which means that all RBridge ports can forward/receive traffic.

- Reduce flooding decapsulated Ethernet frames with unknown MAC-DA to a bridged LAN connected to RBridge edge ports.
When an RBridge receives a TRILL frame whose destination Nickname matches with its own, the normal procedure is for the RBridge to decapsulate the TRILL header and forward the decapsulated Ethernet frame to its directly attached bridged LAN. If the destination MAC is unknown, the decapsulated Ethernet frame is flooded in the LAN. With directory assistance, the RBridge edge can determine if DA in a frame matches with any hosts attached via the bridged LAN. Therefore, frames can be discarded if their DAs do not match.

Reduce the amount of MAC&VLAN <-> RBridgeEdge mapping maintained by RBridge edge. There is no need for an RBridge edge to keep the MAC entries for hosts which don’t communicate with hosts attached to the RBridge edge.

5. Generic operation of Directory Assistance

5.1. Information in Directory Servers for TRILL

To achieve the benefits of directory service for TRILL, the corresponding directory server will need minimum following attributes:

[IP, MAC, attached RBridge nickname, {list of interested RBridges}]

The {list of interested RBridges} would get populated when an RBridge queries for information, or pushed down from management systems. The list is used to notify those RBridges if VMs to RBridge’s connectivity changes due to VMs migration or link failures.

There can be two different models for RBridge edge node to be assisted by Directory Service: Push Model and Pull Model.

5.2. Push Model

Under this model, Directory Server(s) push down the MAC&VLAN <-> RBridgeEdge mapping for all the hosts which might communicate with hosts attached to an RBridge edge node. With this environment, it is recommended that RBridge edge simply drop a data packet (instead of flooding to RBridge domain) if the packet’s destination address can’t be found in the MAC&VLAN<->RBridgeEdge mapping table.
It may not be necessary for every RBridge edge to get the entire mapping table for all the hosts in a data center. There are many ways to narrow the full set down to a smaller set of remote hosts which communicate with hosts attached to an RBridge edge. A simple approach of only pushing down the mapping for the VLANs which have active hosts under an RBridge edge can reduce the number of mapping entries pushed down.

However, it is inevitable that RBridge edge’s MAC&VLAN<->RBridgeEdge mapping table will have more entries than they really need under the Push Model. When hosts attached to one RBridge Edge rarely communicate with hosts attached to different RBridge edges even though they are on the same VLAN, the normal process of RBridge edge’s unknown DA flooding, learning and cache aging would have removed those MAC&VLAN entries from the RBridge’s cache. But it can be difficult for Directory Servers to predict the communication patterns among hosts within one VLAN. Therefore, it is likely that the Directory Servers will push down all the MAC&VLAN entries if there are hosts in the VLAN being attached to the RBridge Edge. This is a major disadvantage of push down model.

In push down model, it is necessary to have a message for RBridge node to request directory server(s) to start pushing down the mapping entries. This message should at least include the number VLANs enabled on the RBridge, so that directory server doesn’t need to push down the entire mapping entries for all the hosts in the data center. RBridge node can use this message to get mapping entries when it is initialized or restarted.

The detailed message format and hand-shake mechanism between RBridge and Directory Server(s) will be described in a separate draft because this draft only focuses on the framework of directory assisted Edge.

When directory pushes down the entire mapping to an edge RBridge for the very first time, there usually are many entries. To minimize the number of entries pushed down, summarization should be considered, e.g. with one edge RBridge Nickname being associated with all attached hosts’ MAC addresses and VLANs as shown below:
Whenever there is any change in MAC&VLAN <-> RBridgeEdge mapping, which can be triggered by hosts being added, moved, or decommissioned, an incremental update can be sent to the RBridge edges which are impacted by the change. Therefore, something like sequence number has to be maintained by directory servers and R Bridges. Detailed mechanisms will be described in a separate draft.

5.3. Pull model:

Under this model, ‘’R Bridge’’ pulls the MAC&VLAN<->RBridgeEdge mapping entry from the directory server when needed. There are several options to trigger the pulling process. For example, the R Bridge edge node can send a pulling request whenever it receives an unknown DA, or R Bridge edge node can simply intercept all ARP/ND requests and forward them to the Directory Server(s) that has the information on where each host is located. R Bridge ingress node can cache the mapping pulled down from the directory.

One advantage of the Pull Model is that R Bridge edge can age out MAC&VLAN entries if they haven’t been used for a certain period of time. Therefore, each R Bridge edge will only keep the entries which are frequently used, i.e. mapping table size can be smaller. R Bridge edge would query the Directory Server(s) for unknown DAs in data frames or ARP/ND and cache the response. When hosts attached to one R Bridge Edge rarely communicate with hosts attached to different R Bridge edges even though they are on the same VLAN, the corresponding MAC&VLAN entries would be aged out from the R Bridge’s cache.
Some people are concerned of the performance with RBridge waiting for response from Directory Servers upon receiving a data frame with unknown DA. Actually this waiting practice is a common router behavior. Most deployed routers today do hold the packets and send an ARP/ND to the target upon receiving a packet with DA not in its IP-MAC cache. When ARP/ND replies are received, the router will send the data frame to the target. This practice is to minimize flooding when targets don’t exist in the subnet.

When the target doesn’t exist in the subnet, routers generally re-send ARP/ND request a few more times before dropping the packets. Therefore, the holding time by routers to wait for ARP/ND response can be longer than the time taken by the Pull Model to get IP-MAC mapping from directory if target doesn’t exist in the subnet.

A separate draft will describe the detailed messages and mechanism for RBridge edge to pull information from directory server(s).

6. Conclusion and Recommendation

The traditional RBridge learning approach of observing data plane can no longer keep pace with the ever growing number of hosts in Data center.

Therefore, we suggest TRILL consider directory assisted approach(es). This draft only describes the basic framework of using directory assisted approach for RBridge edge nodes. More complete mechanisms will be described in separate drafts.

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


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


[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
ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS
FOR A PARTICULAR PURPOSE.

Acknowledgment

Funding for the RFC Editor function is currently provided by the
Internet Society.
Joynt Zhai
INTERNET-DRAFT
Intended status: Proposed Standard
Updates: 6325
Expires: September 11, 2012

TRILL: The ESADI Protocol
<draft-hu-trill-rbridge-esadi-03.txt>

Abstract

The IETF TRILL (TRansparent Interconnection of Lots of Links) protocol provides least cost pair-wise data forwarding without configuration in multi-hop networks with arbitrary topologies and link technologies. TRILL supports the multi-pathing of both unicast and multicast traffic. Devices that implement the TRILL protocol are called RBridges (Routing Bridges) or TRILL Switches.

The ESADI (End System Address Distribution Information) protocol is a VLAN (Virtual Local Area Network) scoped way that TRILL switches can communicate end station addresses to each other. An RBridge announcing VLAN-x connectivity (normally a VLAN-x forwarder) and running the TRILL ESADI protocol can receive remote address information and/or transmit local address information for VLAN-x to other such RBridges. This document updates RFC 6325, specifically the documentation of the ESADI protocol.

Status of This Memo

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

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

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

Acknowledgements

TBD
Table of Contents

1. Introduction............................................4
1.1 Content and Precedence.................................5
1.2 Terminology............................................5

2. ESADI Protocol Overview.................................6
3. ESADI DRB State............................................9

4. ESADI PDU processing...................................10
4.1 Sending of ESASI PDUs.................................10
4.2 Receipt of ESADI PDUs.................................11

5. ESADI-LSP Contents.....................................12
5.1 ESADI Parameter Data..................................12
5.2 MAC Reachability TLV..................................13

6. IANA Considerations....................................14
6.1 ESADI Participation Flag..............................14
6.2 TRILL GENAPP TLV......................................14

7. Security Considerations.................................16

8. References.............................................17
8.1 Normative references..................................17
8.2 Informative References.................................18
1. Introduction

The IETF TRILL (TRansparent Interconnection of Lots of Links) protocol [RFC6325] provides least cost pair-wise data forwarding without configuration in multi-hop networks with arbitrary topologies and link technologies, safe forwarding even during periods of temporary loops, and support for multi-pathing of both unicast and multicast traffic. TRILL accomplishes this by using the IS-IS (Intermediate System to Intermediate System) [IS-IS] [RFC1195] [RFC6326] link state routing protocol and encapsulating traffic using a header that includes a hop count. The design supports VLANs (Virtual Local Area Networks) and optimization of the distribution of multi-destination frames based on VLANs and IP multicast groups. Devices that implement TRILL are called RBridges (Routing Bridges) or TRILL switches.

There are five ways an RBridge can learn end station addresses as described in Section 4.8 of [RFC6325]. The ESADI (End Station Address Distribution Information) protocol is an optional VLAN scoped way RBridges can communicate end station addresses with each other. An RBridge that is announcing connectivity to VLAN-x (normally a VLAN-x appointed forwarder) MAY use the (ESADI) protocol to announce the end station address of some or all of its attached VLAN-x end nodes to other RBridges that are running ESADI for VLAN-x.

By default, RBridges with connected end stations learn addresses from the data plane when ingressing and egressing native frames. The ESADI protocol’s potential advantages over data plane learning include the following:

1. Security advantages: The ESADI protocol can be used to announce end stations with an authenticated enrollment (for example enrollment authenticated by cryptographically based EAP (Extensible Authentication Protocol [RFC3748]) methods via [802.1X]). In addition, the ESADI protocol supports cryptographic authentication of its message payloads for more secure transmission.

2. Fast update advantages: ESADI protocol provides a fast update of end nodes MAC (Media Access Control) addresses. If an end station is unplugged from one RBridge and plugged into another, frames addressed to that older RBridge can be black holed. They can be sent just to the older RBridge that the end station was connected to until cached address information at some remote RBridge times out, possibly for tens of seconds [RFC6325].

MAC address reachability information and some ESADI parameters are carried in ESADI frames rather than in the TRILL IS-IS protocol. As described below, ESADI is, for each VLAN, a virtual logical topology overlay in the TRILL topology. An advantage of using ESADI is that
the end station attachment information is not flooded to all RBridges through TRILL IS-IS but only to participating RBridges advertising ESADI support for the VLAN in which those end stations occur.

1.1 Content and Precedence

This document updates the description of the ESADI protocol in the TRILL basic specification.

Section 2 is the ESADI protocol overview. Section 3 specifies ESADI DRB state. Section 4 discusses the processing of ESADI PDUs. Section 5 describes the ESADI-LSP contents.

This document updates [RFC6325] and prevails over [RFC6325] in the case of conflicts.

1.2 Terminology

This document uses the acronyms defined in [RFC6325] and the following phrase:

LSP number zero - A Link State PDU with fragment number equal to zero.

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. ESADI Protocol Overview

ESADI is a VLAN scoped way that RBridges can announce and learn end station addresses rapidly and securely. An RBridge that is announcing itself as connected to one or more VLANs (usually because it is an Appointed Forwarder) and participates in the ESADI protocol is called an ESADI RBridge.

ESADI is a separate protocol from the TRILL IS-IS implemented by all RBridges in a campus. There is a separate ESADI instance for each VLAN. In essence, for each VLAN, there is a modified instance of the IS-IS reliable flooding mechanism in which ESADI RBridges may choose to participate. (These are not the instances being specified in [MultiInstance].) It is an implementation decision how independent the implementations of multiple ESADI instances at an RBridge are. For example, the ESADI link state could be in a single database with a field in each record indicating the VLAN to which it applies or could be a separate database per VLAN. But the update processes operate separately for each ESADI instance.

After the TRILL header, ESADI frames have an inner Ethernet header with the Inner.MacDA of "All-Egress-RBridges" (formerly called "All-ESADI-RBridges"), an Inner.VLAN tag specifying the VLAN of interest, and the "L2-IS-IS" Ethertype followed by the ESADI payload as shown in Figure 1. For more detail see Section 4.2.5 in the TRILL base protocol specification [RFC6325].

TRILL ESADI frame Structure

```
+x--------------------------------+      +--------------------------------+
|          Link Header            |      |       TRILL Data Header         |
+--------------------------------+      +--------------------------------+
|       Inner Ethernet Header    |      |       Inner Ethernet Header     |
+--------------------------------+      +--------------------------------+
|         ESADI Payload          |      |       ESADI Payload            |
+--------------------------------+      +--------------------------------+
|          Link Trailer          |      |       Link Trailer             |
+--------------------------------+      +--------------------------------+
```

Figure 1

All transit RBridges forward ESADI frames as if they were ordinary multicast TRILL Data frames. Because of this forwarding, it appears to an instance of the ESADI protocol at an RBridge that it is directly connected by a multi-access virtual link to all other RBridges in the campus running ESADI for that VLAN. Thus no "routing" computation or routing decisions ever have to be performed by ESADI. An ESADI RBridge merely transmits the ESADI frames it originates on...
this virtual link as described for any multicast frame in [RFC6325] using any distribution tree that it might use for a normal TRILL Data frame. RBridges that do not implement the ESADI protocol, do not have it enabled, or are not participating for the Inner.VLAN of an ESADI frame do not decapsulate or locally process any TRILL ESADI frames they receive. Thus the ESADI frames are transparently tunneled through transit RBridges.

TRILL ESADI frame payloads are structured like IS-IS PDUs, except as indicated below, but are always TRILL encapsulated on the wire as if they were TRILL Data frames.

The ESADI instance for VLAN-x at an RBridge RB1 determines who its ESADI potential neighbors are by logically examining the TRILL IS-IS link state database for RBridges that are data and IS-IS reachable from RB1 (see Section 2 of [ClearCorrect]) and are announcing their participation in VLAN-x ESADI. When an RBridge RB2 becomes IS-IS or data unreachable from RB1 or any of the relevant entries for RB2 are purged from the core IS-IS link state database, it is lost as a potential neighbor and also dropped from any ESADI instances. And when RB2 is no longer announcing participation in VLAN-x ESADI, it ceases to be a potential neighbor for the VLAN-x ESADI instance. RB2 becomes an actual ESADI adjacency for RB1 when it is a potential neighbor and RB1 holds an ESADI-LSP zero for RB2, all these considerations being VLAN scoped. Because of these mechanisms, there are no "Hellos" sent in ESADI.

The information distributed by the ESADI protocol is a list of local end station MAC addresses connected to the originating RBridge and, for each such address, a one octet unsigned "confidence" rating in the range 0-254 (see Section 5.2). It is entirely up to the originating RBridge which locally connected MAC addresses it wishes to advertise via ESADI and with what confidence. It MAY advertise all, some, or none of such addresses it has. Future uses of ESADI may distribute additional types of information.

TRILL ESADI-LSPs MUST NOT contain a VLAN ID in their payload. The VLAN ID to which the ESADI data applies is the Inner.VLAN of the TRILL Data frame enclosing the ESADI payload. If a VLAN ID could occur within the payload, it might conflict with the Inner.VLAN and could conflict with any future VLAN mapping scheme that may be adopted [VLANmapping]. If a VLAN ID field in an ESADI-LSP PDU does include a VLAN ID, its contents is ignored.

(In the future, TRILL may be extended to provide more fine-grained labeling of data and ports. If so, it is expected that ESADI will be extended by allowing such fine-grained labeling of ESADI frames, as an alternative to the currently allowed Inner.VLAN labeling. As with the current ESADI specification, it would generally be prohibited for such fine-grained labeling information to appear inside such extended
ESADI frames.)
3. ESADI DRB State

Generally speaking, the DRB state on an ESADI link operate similarly to a TRILL IS-IS broadcast link with the following exception:

In the ESADI-DRB election at RB1 on an ESADI link, comparing with [RFC6327], the candidates are the local ESADI instance for VLAN-x and all remote ESADI instances at RBridges that (1) are data and IS-IS reachable from RB1 [ClearCorrect], (2) are announcing in their TRILL IS-IS LSP that they are participating in ESADI for VLAN-x, and (3) for which RB1 is holding an ESADI-LSP zero with an ESADI Parameters APPsub-TLV. The winner is the instance with the highest ESADI Parameter 7-bit priority field with ties broken by System ID, comparing fields as unsigned integers with the larger magnitude considered higher priority. In particular "SNPA/MAC address" is not considered and there is no "Port ID".

Because ESADI does no routing, the ESADI-DRB does not create a pseudo-node.
4. ESADI PDU processing

VLAN-x ESADI neighbors are usually not connected directly by a physical link, but are always logically connected by a virtual link. There could be hundreds of ESADI RBridges on the virtual link. There are only EASDI-LSP, EASDI-CSNP and EASDI-PSNP PDUs used in ESADI. In particular, there are no Hello or MTU PDUs because ESADI does not build a topology and does not do any routing.

In IS-IS, multicasting is normally on a local link and no effort is made to optimize to unicast because under the original conditions when IS-IS was designed (commonly a piece of multi-access Ethernet cable), any frame made the entire link busy for that frame time. But in ESADI what appears to be a simple multi-access link is actually a multi-hop distribution tree that may or may not be pruned. Thus, transmitting a multicast frame on such a tree imposes a greater load than transmitting a unicast frame. This load may be justified if there are likely to be multiple listeners but may not be justified if there is only one recipient of interest. For this reason, under some circumstances and if the target indicates that it has the capability to receive unicast ESDAI, ESADI PDUs MAY be TRILL unicast.

Section 4.1 describes the sending of ESADI PDUs. Section 4.2 covers the receipt of ESADI PDUs.

4.1 Sending of ESADI PDUs

The MTU available to instances of ESADI is 24 bytes less than that available to TRILL IS-IS because of the additional fields required (2(TRILL Ethertype) + 6(TRILL Header) + 6(Inner.MacDA) + 6(Inner.MacSA) + 4(Inner.VLAN)). Thus the inner ESADI payload, starting with the Intradomain Routing Protocol Discriminator, MUST NOT exceed Sz minus 24; however, if a larger payload is received, it is processed normally.

Once an ESADI instance is operationally up, it multicasts it self-originated LSP number zero on the virtual link to announce its ESADI parameters. When the other ESADI instances receive the LSP number zero and find a new neighbor, their self-originated LSP fragments are scheduled to be sent and MAY be unicast to that neighbor. If all the other ESADI instances send their self-originated LSPs immediately, there may be a surge of traffic to that new neighbor. So the other ESADI instances should wait an interval time before sending the LSP to a new neighbor. The interval time value is up to the device implementation. One suggestion is that the interval time can be assigned a random value with a range based on the ESADI priority when implementation.
If the ESADI instance is DRB, it multicasts an ESADI-CSNP periodically to keep the Link State Database synchronized among its neighbors on the virtual link. After receiving an ESADI-PSNP PDU, the DRB will transmit the LSPs requested by the PSNP on the virtual link.

For robustness, if an ESADI instance has two or more ESADI neighbors and is not DRB and it receives no ESADI-CSNP PDUs for at least the CSNP Time (see Section 5.1) of the DRB, it MAY transmit an ESADI-CSNP.

In the case of receiving an ESADI-LSP with a smaller sequence number than the copy stored in local EASDI Link State Database, the local ESADI instance will also schedule to transmit the stored copy and MAY unicast it to the sender.

The format of a unicast ESADI frame is the format of TRILL ESADI frame, in section 4.2 in [RFC6325], except that, in the TRILL header, the M bit is set to zero and the Egress Nickname is the nickname of the destination RBridge.

4.2 Receipt of ESADI PDUs

Because ESADI adjacency is in terms of System ID, all PDU acceptance tests that check that the PDU is from an adjacent system check that the System ID is that of an ESADI neighbor and do not check the source SNPA/MAC.

Because all ESADI instance for VLAN-x are adjacent, when RB1 receives an ESADI-CSNP from RB2 and detects that it has ESADI-LSPs that RB1 is missing, it sets the transmission flag only for its own EASDI-LSPs that RB1 is missing. Missing EASDI-LSPs originated by other ESADI instances will be detected by those other instances.

When receiving an ESADI-PSNP PDU, if the local ESADI instance is DRB, ESADI-LSP PDU requested by the ESADI-PSNP will be multicast on the virtual link.
5. ESADI-LSP Contents

The only PDUs used in ESADI are the Level 1 ESADI-LSP, ESADI-CSNP, and ESADI-PSNP PDUs. The content of an ESADI-LSP consists of zero or more MAC Reachability TLVs, optionally an Authentication TLV, and exactly one ESADI parameter APPsub-TLV in ESADI-LSP zero. This section specifies the format for ESADI parameter data APPsub-TLV and gives the reference for the ESADI MAC Reachability TLV. In the future, there may be other TLVs or sub-TLVs carried in EASDAI-LSPs.

ESADI-LSP number zero MUST NOT exceed 1470 minus 24 bytes in length (1446 bytes) but if received longer, it is still processed normally.

5.1 ESADI Parameter Data

The figure below presents the format of the ESADI parameter data. This APPsub-TLV MUST be included in a TRILL GENAPP TLV in ESADI-LSP number zero. If it is missing, priority is assumed to be zero and CSNP time 40. If there is more than one occurrence, the first occurrence will be used.

```
+--------+-
| Type    |
+--------+- (1 byte)

+--------+-
| Length  |
+--------+- (1 byte)

+--------+-
| R Priority |
+--------+- (1 byte)

+--------+-
| CSNP Time |
+--------+- (1 byte)

+--------+-
| Reserved for expansion |
+--------+- (variable)
```

Type: set to TRILL APPsub-TLV type 1.

Length: Set to 2 to 255.

R: A reserved bit that MUST be sent as zero and ignored on receipt.

Priority: The Priority field gives the ESADI instance’s priority for being DRB on the TRILL ESADI virtual link for the VLAN in which the PDU containing the parameter data was sent. It is an unsigned seven-bit integer with larger magnitude indication higher priority. It defaults to 0x40.

CSNP Time: An unsigned byte that give the amount of time in seconds during the originating RBridge, if it is DRB on the ESADI link,
will send at least 3 EASDI-CSNP PDUs. It defaults to 30 seconds.

Reserved for future expansion: Future versions of the ESADI Parameters APPsub-TLV may have additional information. A receiving ESADI RBridge ignores any additional data here unless it implements such future expansion(s).

5.2 MAC Reachability TLV

The information in TRILL ESADI-LSP PDUs consists of one or more MAC Reachability (MAC-RI) TLVs as specified in [RFC6165]. These TLVs contain one or more unicast MAC addresses of end stations that are both on a port and in a VLAN for which the originating RBridge is appointed forwarder, along with the one octet unsigned Confidence in this information with a value in the range 0-254. If such a TLV is received with a confidence of 255, it is treated as if the confidence was 254.

To avoid conflict with the Inner.VLAN ID, the TLVs in TRILL ESADI PDUs, including the MAC-RI TLV, MUST NOT contain the VLAN ID. If a VLAN-ID is present in the MAC-RI TLV, it is ignored. The VLAN to which the ESADI-LSP applies is indicated only by the Inner.VLAN tag in the encapsulated TRILL ESADI frame.
6. IANA Considerations

IANA allocation considerations are given below.

6.1 ESADI Participation Flag

IANA is requested to allocate an "ESADI Participation" bit in the Interested VLANs and Spanning Tree Roots sub-TLV [RFC6326]. (bit 2 in the Interested VLANs field recommended) If this bit is a one, it indicates that the originating RBridge is participating in ESADI for the indicated VLAN or VLANs.

6.2 TRILL GENAPP TLV

IANA is requested to allocate an IS-IS Application Identifier under the Generic Information TLV (#251) for TRILL [RFCgenapp] and to create a subregistry in the TRILL Parameters Registry for "TRILL APPsub-TLVs under IS-IS TLV #251 Application Identifier #TBD". The initial contents of this subregistry are as follows:

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
<td>&lt;this RFC&gt;</td>
</tr>
<tr>
<td>1</td>
<td>ESADI Parameters</td>
<td>&lt;this RFC&gt;</td>
</tr>
<tr>
<td>2-254</td>
<td>Available</td>
<td>&lt;this RFC&gt;</td>
</tr>
<tr>
<td>255</td>
<td>Reserved</td>
<td>&lt;this RFC&gt;</td>
</tr>
</tbody>
</table>

TRILL APPsub-TLV Types 2 through 254 are available for allocation by Standard Action, as modified by [RFC4020]. For example, such APPsub-TLVs might be used in connection with OAM [OAMdraft]. The standards track RFC causing such an allocation will also include a discussion of security issues and of the rate of change of the information being advertised. TRILL APPsub-TLVs MUST NOT alter basic TRILL IS-IS protocol operation including the establishment of adjacencies, the update process, and the decision process for TRILL IS-IS [IS-IS] [RFC1195] [RFC6327]. The TRILL Generic Information TLV MUST NOT be used in IS-IS instance zero.

The V, I, D, and S flags in the initial flags byte of a TRILL Generic Information TLV have the meanings specified in [RFCgenapp] but are not currently used as TRILL operates as a Level 1 IS-IS area and no semantics is hereby assigned to the inclusion of an IPv4 and/or IPv6 address via the I and V flags. Thus these flags MUST be zero; however, use of multi-level IS-IS is an obvious extension for TRILL
[MultiLevel] and future IETF Standards Actions may update or obsolete this specification to provide for the use of any or all of these flags in the TRILL GENAPP TLV.

The ESADI Parameters information, for which APPsub-TLV 1 is hereby assigned, is compact and slow changing (see Section 5.1).

For Security Considerations related to ESADI and the ESADI parameters APPsub-TLV, see Section 7.
7. Security Considerations

For general TRILL Security Considerations, see [RFC6325].

More TBD
8. References

Normative and informative references for this document are below.

8.1 Normative references

[IS-IS] - International Organization for Standardization,
"Intermediate system to Intermediate system intra-domain
routeing information exchange protocol for use in conjunction
with the protocol for providing the connectionless-mode Network
2002.

[RFC1195] - Callon, R., "Use of OSI IS-IS for routing in TCP/IP and

[RFC2119] - Bradner, S., "Key words for use in RFCs to Indicate

[RFC4020] - Kompella, K. and A. Zinin, "Early IANA Allocation of


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

[RFC6326] - Eastlake, D., Banerjee, A., Dutt, D., Perlman, R., and A.
Ghanwani, "Transparent Interconnection of Lots of Links (TRILL)
Use of IS-IS", RFC 6326, July 2011.

[RFC6327] - Eastlake 3rd, D., Perlman, R., Ghanwani, A., Dutt, D.,
and V. Manral, "Routing Bridges (RBridges): Adjacency", RFC
6327, July 2011.

Generic Information in IS-IS", draft-ietf-isis-genapp-04.txt,
in RFC Editor’s queue.

8.2 Informative References


[OAMdraft] - draft-tissa-trill-oam, work in progress.


Authors’ Addresses

Hongjun Zhai
ZTE Corporation
68 Zijinghua Road
Nanjing 200012 China
Phone: +86-25-52877345
Email: zhai.hongjun@zte.com.cn

Fangwei Hu
ZTE Corporation
889 Bibo Road
Shanghai 201203 China
Phone: +86-21-68896273
Email: hu.fangwei@zte.com.cn

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

Donald Eastlake
Huawei R&D USA
155 Beaver Street
Milford, MA 01757 USA
Phone: +1-508-333-2270
Email: d3e3e3@gmail.com
Copyright and IPR Provisions

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License. The definitive version of an IETF Document is that published by, or under the auspices of, the IETF. Versions of IETF Documents that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of IETF Documents. The definitive version of these Legal Provisions is that published by, or under the auspices of, the IETF. Versions of these Legal Provisions that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of these Legal Provisions. For the avoidance of doubt, each Contributor to the IETF Standards Process licenses each Contribution that he or she makes as part of the IETF Standards Process to the IETF Trust pursuant to the provisions of RFC 5378. No language to the contrary, or terms, conditions or rights that differ from or are inconsistent with the rights and licenses granted under RFC 5378, shall have any effect and shall be null and void, whether published or posted by such Contributor, or included with or in such Contribution.
Routing Bridges (RBridges): Operations, Administration, and Maintenance (OAM) Support
draft-ietf-trill-rbridge-oam-02

Abstract

Routing Bridges (RBridges) implement the TRansparent Interconnection of Lots of Links (TRILL) protocol which provide a transparent least-cost frame routing in multi-hop networks with arbitrary topologies, while also inherently providing loop mitigation. As RBridges are deployed in real-world situations, operators will need tools for debugging problems that arise. This document specifies a set of RBridge features for operations, administration, and maintenance (OAM) purposes in RBridge campuses. The features specified in this document include tools for traceroute, ping, and error reporting.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 13, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of
Table of Contents

1. Introduction ......................................................... 4  
   1.1. Requirements Language ........................................ 4  
2. Acronyms ........................................................... 4  
3. TRILL OAM Message .................................................. 6  
4. RBridge Tools ...................................................... 7  
   4.1. Application RBridge Tools ...................................... 7  
      4.1.1. RBridge Ping ............................................. 8  
      4.1.2. Hop Count Traceroute ..................................... 9  
      4.1.2.1. Path Sharing ......................................... 10  
      4.1.2.2. Multi-Destination Targets ............................. 11  
   4.2. Error Reporting ................................................ 12  
      4.2.1. Hop Count Zero Error .................................... 12  
      4.2.2. MTU Error ............................................... 13  
5. RBridge Channel Message Format .................................... 13  
   5.1. RBridge Channel Header and Sequence Number .................. 13  
6. OAM Protocol Field Values .......................................... 14  
   6.1. Response Frame Field Values .................................. 14  
   6.2. Self-Initiated Frame Field Values ............................ 16  
7. OAM Protocol Formats .............................................. 17  
   7.1. Protocol Application Codes Formats .......................... 17  
      7.1.1. Echo Request ............................................ 17  
      7.1.2. Echo Reply .............................................. 18  
   7.2. Error Notification Format .................................... 19  
      7.2.1. Error Specifiers ....................................... 20  
8. Type, Length, Value (TLV) Encodings ................................ 21  
   8.1. Next Hop Information ......................................... 23  
   8.2. Previous Hop Information ..................................... 24  
   8.3. Incoming Port ID ............................................... 24  
   8.4. Outgoing Port ID ............................................. 25  
   8.5. Outgoing Port MTU ............................................ 25  
   8.6. IS-IS System ID ............................................... 26  
9. Acknowledgments .................................................... 26  
10. IANA Considerations ................................................ 26  
11. Security Considerations ........................................... 27  
12. References ........................................................ 27  
   12.1. Normative References ....................................... 27  
   12.2. Informative References ..................................... 28  
Appendix A. Implementation Considerations ................................ 29  
   A.1. Hop Count Traceroute Example ................................ 29  
   A.2. Ping Example .................................................. 31  
Appendix B. Revision History .......................................... 32  
   B.1. Changes from -01 to -02 .................................... 32  
   B.2. Changes from -00 to -01 .................................... 33
1. Introduction

The IETF has standardized RBridges, devices that implement the TRILL protocol, a solution for transparent least-cost frame routing in multi-hop networks with arbitrary topologies, using a link-state routing protocol technology and encapsulation with a hop-count [RFC6325]. As RBridges are deployed, operators will require tools for troubleshooting of operations issues in the network. TRILL uses IS-IS for the control plane [IS-IS] [RFC6165] [RFC6326]. IS-IS has a link-state database which contains the information of all links in the TRILL domain and IS-IS has a routing table. This information can be used for trouble shooting purposes.

There are a number of mechanisms to verify the control plane/data plane information, however correctness of the control plane information does not guarantee the data plane is working correctly. This motivates the need for OAM tools that allow an operator to test the data plane. Protocols such as IP, MPLS, and IEEE 802.1 have features enabling an operator to exercise the data plane [RFC4443] [RFC0792] [IEEE.802-1ag]. There is a need for a similar set of tools in TRILL. Likewise, there is a need for error reporting capabilities inside an RBridge campus.

Sometimes there may be a need for faster convergence than is provided by the TRILL hello protocol. Such fault notification functionality is not specified in this document. [BFD] provides this functionality using BFD.

This document specifies a set of RBridge features for operations, administration, and maintenance purposes in RBridge campuses along with the procedures and frame formats for these features. The features specified in this document include tools for traceroute, ping, and error reporting. Section 3 of this document specifies the general usage of a defined message format. Section 4 specifies some additional applications of the message format. Section 5 specifies the format and value of the messages on the wire.

1.1. Requirements Language

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

2. Acronyms

- BPDU - Bridge PDU
o CHbH - Critical Hop-by-Hop
o CItE - Critical Ingress-to-Egress
o DA - Destination Address
o DR - Designated Router
o DRB - Designated RBridge
o ES - End Station
o ESa - End Station A
o ESb - End Station B
o ECMP - Equal-Cost Multi-Path
o ESADI - End Station Address Distribution Instance
o FCS - Frame Check Sequence
o ID - Identification
o IEEE - Institute of Electrical and Electronics Engineers
o IETF - Internet Engineering Task Force
o IP - Internet Protocol
o IS-IS - Intermediate System to Intermediate System
o MAC - Media Access Control
o MPLS - Multiprotocol Label Switching
o MTU - Maximum Transmission Unit
o OAM - Operations, Administration, and Maintenance
o P2P - Point-to-point
o PDU - Protocol Data Unit
o RBridge - Routing Bridge
o SA - Source Address
To facilitate message passing as needed by the OAM requirements, the TRILL RBridge Channel facility [RBridgeChannel] is utilized.

There are two types of TRILL OAM messages defined in this document carried within an RBridge Channel: application and error notification. Frames with an error notification MUST NOT be generated in response to frames that are an error notification. Implementations SHOULD rate limit the origination of error notifications. Whereas unknown unicast frames are sent as multi-destination messages, sending unknown unicast frames with an error can lead to an amplification attack. As such special care and rate limiting are necessary for error notifications.

Error notification messages contain the error-causing frame or the initial part thereof after its OAM message. The following are two figures showing application and error notification message structure. Section 5 goes into the details of these formats.
Figure 1

```
+---------------------------------------+
|           Outer Link Header           |
+---------------------------------------+
|              TRILL Header             |
+---------------------------------------+
|         Inner Ethernet Header         |
+---------------------------------------+
|        RBridge Channel Header         |
+---------------------------------------+
|     OAM Protocol Specific Payload     |
+---------------------------------------+
|   Offending Frame TRILL Header        |
+---------------------------------------+
|     Offending Frame Inner Link Header |
+---------------------------------------+
| Truncated Offending Frame Payload     |
+---------------------------------------+
```

Error Notification Frame

Figure 2

RBridge campuses do not, in general, guarantee lossless transport of frames so a frame containing a TRILL OAM Message, possibly generated in response to some other frame, might be lost.

4. RBridge Tools

This section specifies a number of RBridge OAM tools. For classification purposes they are divided into two sections, applications and error tools. Both of these tools use messages called echo requests and echo replies. The format is described in Section 5. An echo request is a message that says please respond. The echo reply is that response. The exact usage is further defined in this section. These messages also contain TLV fields which carry additional information in regards to the message. The formats of these TLVs are described in Section 8.

4.1. Application RBridge Tools
4.1.1. RBridge Ping

Ping is a tool for verifying RBridge connectivity. The ping-originating RBridge transmits one or more TRILL data frames with a TRILL OAM message. This message contains the code of an echo request (See Section 7.1.1). The ingress RBridge MUST be the frame-originating RBridge. The egress RBridge is the destination RBridge to which connectivity will be checked. The M bit MUST be zero.

The purpose of the ping is to confirm connectivity of the data plane, and options defined in future drafts MAY be included. The purpose of allowing the addition of options is so that the frame mimics a data frame that follows the same path through the data plane that a ‘real’ data frame would. An RBridge Ping, however, uses the OAM Channel and so depending on the ECMP hashing used by RBridges in the campus it may not in fact share the same path as ‘real’ data going through the network. The traceroute tool has a way to ensure the data follows the same path as the data does and if an operator wishes to test that path the data takes, the traceroute functionality ought to be used.

The echo request MAY have an arbitrary 28-bit unsigned integer sequence number to assist in matching reply messages to the request. In most circumstances, a single echo request is needed to complete the ping but it might be desirable for a single RBridge to ping multiple egress RBridges, or trace differing flows simultaneously. Assigning differing sequence numbers to each frame aids in matching which trace the reply belongs to.


RBridges implementing ping SHOULD issue a reply in response to this request. See Section 11 for reasons that RBridges are allowed to choose not to respond to a request. If an RBridge chooses to respond to the request, the reply MUST consist of one TRILL data frame per request with an OAM message containing the protocol code of an echo reply. The echo reply MUST have the same sequence number as the request being matched.

For the echo reply the ingress RBridge field MUST be the reply-originating RBridge’s nickname. The egress RBridge MUST be the request-originating RBridge’s nickname. The Inner.VLAN, Inner.MacSA, and Inner.MacDA SHOULD default to the values specified in Section 6.1. The M bit MUST be zero for a known unicast ping.

The reply-originating RBridge SHOULD include its 16-bit port ID from the port on which the request was received in the incoming port field of the reply. It SHOULD also include its 16-bit port ID from the
port on which the frame would be forwarded. A port ID of 0xFFFF indicates the frame would not have been forwarded and that the frame was consumed by the RBridge itself.

The reply frame need not follow the same path though the campus as the request. The reply messages are not meant to test the data plane.

End stations are not involved in this the ping process. RBridge pings are from RBridge to RBridge. While the frames sent may emulate data sent from ESa to ESb, the end stations are not, in fact, involved.

The transmitting RBridge MUST wait for a reply frame until a time-out occurs. At that time, the RBridge SHOULD assume the frame was lost, and this SHOULD be indicated to the operator. The length of this time-out is beyond the scope of this document.

4.1.2. Hop Count Traceroute

The ability to trace the path the data takes through the network is an invaluable debugging tool. RBridge traceroute provides this functionality through use of the TRILL OAM message (See Section 3). In a hop-count traceroute, the originating RBridge starts by transmitting one TRILL data frame with a TRILL OAM message. This message contains a protocol code of an echo request (See Section 7.1.1). The ingress RBridge MUST be the RBridge originating the frame.

When a traceroute is initiated, it is either targeting a known unicast target or a multi-destination target as specified by the operator. If the hop-count traceroute is for a known unicast target, the egress RBridge is the destination RBridge to which connectivity will be checked and the M bit MUST be zero. Otherwise, if the hop-count traceroute is for a multi-destination target, the egress RBridge is the distribution tree nickname for the traceroute. Multi-destination targets are handled the same as known unicast targets but require a small amount of additional logic as specified in Section 4.1.2.2.

The first echo request frame transmitted MUST have a hop-count of zero. The RBridge will continue transmitting these echo requests, incrementing the hop-count by one each time until a hop-count error notification from the destination nickname as its ingress nickname is received. If a transit RBridge decrements the hop-count by more than one it MAY transmit multiple hop-count error notifications.

The purpose of the traceroute is to confirm connectivity of the data
plane, and therefore options defined in future drafts MAY be included. The purpose of allowing the addition of options is so that the frame mimics a data frame that follows the same path through the data plane that a ‘real’ data frame would. The ability to share the same path as ‘real’ data is further specified in Section 4.1.2.1.

The echo request MAY have an arbitrary 28-bit unsigned integer sequence number to assist in matching reply messages to the request. This is important for the hop-count traceroute since replies may return to the ingress RBridge in a different order then their matching requests were sent.


The replying RBridge SHOULD include its 16-bit port ID from the port on which the hop-count error generating frame was received in the Incoming Port ID TLV of the reply. It SHOULD also include its 16-bit port ID from the port on which the frame would be forwarded if the frame did not have a hop-count error in the Outgoing Port ID TLV. A port ID of 0xFFFF indicates the frame would not have been forwarded and would be consumed by the RBridge itself. Finally the reply SHOULD include a 16-bit nickname and 48-bit system id of the next hop RBridge the frame would have been sent to if there were no error in the Next Hop Nickname TLV. If this RBridge is the egress RBridge this TLV MUST NOT be included in the response. This is to facilitate knowledge of a more precise path through the campus as seen in RFC 5837 [RFC5837].

The advantage of this traceroute method is that the transit RBridges do not have to do any special processing of the frames until a hop-count error is detected, a condition they are required to detect by the TRILL base protocol. The disadvantage is the request-originating RBridge needs to transmit as many frames as there are hops between itself and the destination RBridge.

The end stations are not involved in this process. RBridge traceroutes are from RBridge to RBridge. While the frames sent may emulate data sent from ESa to ESb, the end stations are not, in fact, involved. An Rbridge must keep the TRILL header contents the same for ever frame sent in a hop count traceroute.

4.1.2.1. Path Sharing

In certain cases it could be important to send ‘real’ data over a network as to test the path that ‘real’ data takes and to test the fate that such real data would have. Simple sending an RBridge channel message is insufficient because many RBridge implementations
will use various forms of ECMP hashing based on fields such as MAC addresses, IP addresses, and/or TCP/UDP port numbers. To satisfy this need for path sharing an RBridge originating a traceroute MAY send a data packet instead of an echo request. The data packet will look entirely like an encapsulated data frame, with whatever fields the user specifies to ensure path sharing. The one exception is that the hop count will be set as described previously: incremented as the traceroute proceeds. Since these frames will not include a sequence number, these data frames must be sent in lock step: waiting for a timeout or an hop count error before sending the next incremented hop count frame. Since this data frame looks like a real frame but is in fact not real, when the egress RBridge is reached in the traceroute the originating RBridge MUST NOT send trace frames with higher hop-counts. RBridge ping does not have an equivalent path sharing mechanism since it tests end to end connectivity rather than the exact path taken.

4.1.2.2. Multi-Destination Targets

For multi-destination targets at each branch in the tree the tagged frame will be replicated causing each RBridge in the tree, possibly pruned by VLAN and/or IP multicast group, to send a response to the echo request. If all RBridges in the possibly pruned distribution tree support the echo request message, then the ingressing RBridge will receive an error notification from each of them. These error replies are staggered by distance from the generating RBridge. Meaning the first set of responses come from the first request send with hop count equal to zero and these replies will be from this RBridges neighbors. The second set of responses will come from RBridge two hops away and so forth.

The ingressing RBridge can compile all of these notifications, using the parent pointers located in the previous hop information TLV, into an output of the tree the traffic traversed. A traceroute application SHOULD report any errors received, such as an invalid distribution tree nickname, caused by the hop-count traceroute frames. RBridges receiving a multicast destination echo request MUST NOT transmit an echo reply if the multi-destination bit is set. Echo requests that are not used with the hop-count traceroute come from the ping tool, and ping messages are not valid as multi-destination traffic. In a hop count traceroute, devices will already be transmitting a hop-count error notification and so there is no reason to transmit a double set of replies. A multi-destination hop-count traceroute stops when the transmitted hop count reaches the maximum, 0x3F. One cannot use the diameter of the network to limit when this traceroute stops because some RBridges may decrement the hop count by more than one.
In multi-destination request frames, the Previous Information TLV MUST be set to the nickname and system id of the RBridge the frame was received from. This is the previous hop RBridge. The Next Hop Information TLV is not used in multi-destination traceroute frames.

4.2. Error Reporting

Errors can occur in received TRILL data frames. For this purpose, the error notification format is specified. These are generated due to various events as specified subsequently. When a TRILL data frame is received with an error, an error notification frame SHOULD be generated. See Section 11 for reasons some RBridges are allowed to choose not to respond to a request. The generated reply MUST contain the error notification. The sub-code MUST contain a code specifying the error encountered. The valid sub-code values are specified in Section 7.2.1. Two of these sub-codes provide for TLVs with additional information. The error notification also contains a 3 bit error type field which describes the error.

This frame has a TRILL header and it contains, as its payload, the frame received with the error. If the size of the received frame would cause the generated frame to exceed 1470 bytes, the frame MUST be truncated to the 1470 bytes. The payload MUST include the TRILL header of the received frame and MUST NOT include the link-layer header. The generated reply MUST contain the error notification message specific to the error.

When the original ingress RBridge receives the error frame, at a minimum, the RBridge SHOULD update a counter specifying the number of error frames received for the causing error. The encapsulated frame MUST NOT be egressed.

The two sub-codes that provide for TLVs with additional information are described below. All other sub-codes specified in this document do not normally contain TLVs.

4.2.1. Hop Count Zero Error

When a TRILL data frame is received with a hop-count of zero, an error notification frame SHOULD be generated unless rate limiting or some particular difficulty, as described below, stops the sending of such an error notification. The generated reply MUST contain the hop-count zero error sub-code. If the received frame has the echo request message, the hop-count zero error notification MUST have a sequence number matching the echo request. Otherwise, the sequence number MUST be set to zero. The Incoming Port ID TLV SHOULD be included with the port ID the received frame arrived on. The Outgoing Port ID TLV SHOULD be included with the port ID of the port
the received frame would have been forwarded onto if the hop-count was not zero. Finally, the error notification SHOULD include a 16-bit nickname and 48-bit system id of the next hop RBridge the frame would have been sent to in the Next Hop Information TLV. If the request is a multi-destination frame, the previous hop information SHOULD be included instead with it set to the nickname and system id of the RBridge the frame was received from. This is the previous hop RBridge. If the RBridge transmitting the reply is the egress RBridge, this TLV MUST NOT be included in the frame.

4.2.2. MTU Error

When a TRILL data frame is received with a payload that would exceed the MTU of the port the frame would otherwise be forwarded to, an error notification frame MAY be generated. The generated reply MUST contain the MTU error sub-code. The Outgoing Port MTU TLV MUST be included with the MTU of the port the frame would have otherwise been transmitted on. The Incoming Port ID TLV SHOULD be included with the port ID the received frame arrived on. The Outgoing Port ID TLV SHOULD be included with the port ID of the port the received frame would have been forwarded onto if the frame size was not too large. Finally, the error notification message SHOULD include a 16-bit nickname and 48-bit system id of the next hop RBridge the frame would have been sent to in the Next Hop Information TLV. If the request is a multi-destination frame, the previous hop information SHOULD be included instead with it set to the nickname and system id of the RBridge the frame was received from. This is the previous hop RBridge. If the RBridge transmitting the reply is the egress RBridge, this TLV MUST NOT be included in the frame.

5. RBridge Channel Message Format

This section specifies the format of the TRILL OAM payload after the RBridge Channel header and values of the fields in the RBridge Channel Header [RBridgeChannel].

5.1. RBridge Channel Header and Sequence Number

The RBridge Channel Header [RBridgeChannel] fields and flags and following sequence number are as follows:

- CHV (Channel Header Version) is zero.
- Protocol code values are:
  * 0x004 (Suggested): Echo
* 0x005 (Suggested): Error Notification

- Flags: The SL and NA bits SHOULD be zero, the MH bit SHOULD be one
- ERR: The ERR field MUST be zero.

- SPID and Sequence Number: For the Echo and Error Notification protocols, the RBridge Channel Header is always followed by a nibble sub-protocol identifier (SPID) and a 28-bit Sequence Number. This 28-bit field is used to sequence or match frames for certain uses. The SPID is used to provide additional op-code room for a protocol to further multiplex its messages. Not all TRILL OAM messages utilize the sequence number field or the SPID.

6. OAM Protocol Field Values

6.1. Response Frame Field Values

Frames with a TRILL OAM message generated in response to another TRILL data frame have fields set as follows unless otherwise specified:

- Frames of type Application or Error
  - Field: Inner.MacSA
    * Value: If the Inner.MacDA of the received frame is one of the MAC addresses of the RBridge generating the frame, the value MUST be that MAC address. Otherwise, it MUST be one of the RBridge’s MAC addresses.

- Frames of type Application or Error
  - Field: Inner.MacDA
    * Value: The value MUST be All-Egress-RBridges.

- Frames of type Application or Error
  - Field: Inner.VLAN ID
    * Value: If the frame is generated in response to another frame with a legal Inner.VLAN ID, it MUST be the Inner.VLAN ID of the received frame. In other cases, it SHOULD be the default VLAN ID 1.
Frames of type Application or Error

* Field: Ingress RBridge nickname
* Value: If the egress RBridge nickname of the received frame is a nickname of the RBridge generating the frame, then the value MUST be that nickname. otherwise, it MUST be one of the RBridge’s nicknames.

Frames of type Application or Error

* Field: Egress RBridge nickname
* Value: The value MUST be the ingress RBridge nickname of the received frame. Except that, if the ingress RBridge nickname received is unknown or reserved the frame MUST be generated on the port the frame was received on with an Outer.MacDA and egress RBridge nickname of the previous-hop RBridge if this is known.

Frames of type Error

* Field: Offending Encapsulated Frame
* Value: The value MUST be N bytes of the frame that had the error where N is the minimum of the frame size and the number of bytes that would bring the resulting error frame up to 1470 bytes. This MUST include the TRILL header and MUST NOT include the link-layer header.

Frames of type Application

* Field: M Bit
* Value: The value of this field is defined by each specific OAM protocol.

Frames of type Error

* Field: M Bit
* Value: The value MUST be zero.

Frames of type Application or Error

* Field: Inner.Priority
* Value: The value SHOULD be one less than the priority of the received frame, but not less than the lowest priority. One less may be numerically one less in the normal case or logically one less in the case of priority mapping being present.

6.2. Self-Initiated Frame Field Values

Frames with a TRILL OAM message that are self-initiated have fields set as follows unless otherwise specified:

- Frames of type Application
  - Field: Inner.MacSA
    * Value: This SHOULD be one of the transmitting RBridge’s MAC addresses. The Inner.MacSA MAY be other values as specified in Appendix A.

- Frames of type Application
  - Field: Inner.MacDA
    * Value: The value SHOULD be All-Egress-RBridges.

- Frames of type Application
  - Field: Inner.VLAN ID
    * Value: The value SHOULD be the default VLAN ID 1. The Inner.VLAN ID MAY be other values as specified in Appendix A.

- Frames of type Application
  - Field: Ingress RBridge nickname
    * Value: The value SHOULD be one of the RBridge’s nicknames. The Ingress RBridge nickname MAY be other values as specified in Appendix A.

- Frames of type Application
  - Field: Egress RBridge nickname
    * Value: The value of this field is defined by each specific OAM protocol.
Frames of type Application

* Field: M Bit
  * Value: The value of this field is defined by each specific OAM protocol.

Frames of type Application

* Field: Inner.Priority
  * Value: The value of this field defaults to zero. The Inner.Priority MAY be other values as specified in Appendix A.

7. OAM Protocol Formats

The formats of Echo Request, Echo Reply, and Error Notification OAM Messages are given below.

7.1. Protocol Application Codes Formats

7.1.1. Echo Request

```
+--------+--------+
| 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 |
+--------+--------+
|        |        |
| RBridge Channel | Header |
|        |        |
|        |        |
|        |        |
| SPID   | Sequence Number |
|        |                 |
+--------+--------+
```

Echo Request

Figure 3

This message is used by ingress RBridges to request an echo reply from the egress RBridge. Further uses are specified in Section 4.1.2 and Section 4.1.1.

* SPID: The SPID MUST be zero to indicate an echo request.

* Sequence Number: An arbitrary 28-bit unsigned integer used to aid in matching reply messages to echo requests. It MAY be zero.
7.1.2. Echo Reply

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|11|12|13|14|15 |
|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| +------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| RBridge Channel  | Header           | SPID             | Sequence         | Number           | TLVs             |
| +------------------|------------------|------------------|------------------|------------------|------------------|
| +------------------|------------------|------------------|------------------|------------------|------------------|

Echo Reply Format

Figure 4

This message is used by egress RBridges to reply to an echo request from the ingress RBridge. Further uses are specified in Section 4.1.2 and Section 4.1.1.

- **SPID**: The SPID MUST be one to indicate an echo reply.

- **Sequence Number**: A 28-bit unsigned integer used to aid in matching reply messages to echo requests. Set to the sequence number field of the Echo Request that cause this Echo Reply.

- **TLVs**: A set of type, length, value encoded fields as specified in Section 8.
7.2. Error Notification Format

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>+-----------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RBridge Channel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Header</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPID</td>
<td>Sequence Number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Err. T.</td>
<td>Subcode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>TLVs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Error Format**

*Figure 5*

This message is used by RBridges to signal that an error has been detected.

- **SPID**: The SPID MUST be set to all zeros on transmission and is ignored on reception. It is unused by the error notification protocol.

- **Sequence Number**: For all sub-codes except for the hop count error this field is unused. It is set to zero on transmission and ignored on reception. For the hop count error this is a 28-bit unsigned integer used to aid in matching reply messages to echo requests. If the frame whose hop-count dropped to zero contains the echo request message (See Section 7.1.1), this MUST match the sequence number Echo Request found in that message. If this is not in reply to an Echo Request, then the sequence number MUST be set to zero.

- **Error Type**: MUST be a specifier of the error type describing the error. The values are: 0 (Permanent Error), 1 (Transient Error), 2 (Warning), 3 (Comment). Values 4 through 7 are available for allocation by IETF Review.

- **Subcode**: MUST be a specifier of the error discovered in the frame. The valid values are specified in Section 7.2.1
7.2.1. Error Specifiers

The sub-code values fall into three categories: errors (divided into transient and permanent errors), warnings, and comments. All sub-codes represent something out of the ordinary that has gone wrong, but certain ones are more important than others. Sub-codes that are classified as errors are the most severe with warning sub-codes being less severe. These are enabled by default. Errors can be further divided into transient and permanent. Transient errors are errors that happen but where the error causing RBridge can try again in the future and the error may not happen again. Permanent errors are errors that will happen again in a converged network. It is up to implementations to determine if errors should be listed as permanent or transient. Sub-codes classified as comments are minor and are disabled by default. They may be useful for operators debugging a network. All error generations are optional and therefore MAY be generated or not generated depending on security and implementation constraints.

The error specifiers sub-code values are:

Error Sub-codes

- 0: Unknown Error: Indicates an error has occurred.
- 1: Invalid Outer.VLAN: Indicates the Outer.VLAN ID was not the designated VLAN ID or was 0xFFFF.
- 2: Unknown Egress RBridge: Indicates the Egress RBridge in a received frame is unknown.
- 3: Unknown Ingress RBridge: Indicates the Ingress RBridge in a received frame is unknown. (RBridges are not required to test for this error.)
- 4: Unsupported Critical Hop-by-hop Option: Indicates an unsupported critical hop-by-hop option was received.
- 5: Unsupported Critical Ingress-to-Egress Option: Indicates an unsupported critical ingress-to-egress option was received.
- 6: Hop Count Zero: Indicates a frame hop count reached zero in transit. (Used for pings and traceroute.)
8. Type, Length, Value (TLV) Encodings

To facilitate future interoperable expansion of the data carried in OAM sub-messages some sub-messages use a TLV encoding. These TLV sections consist of a list of type, length, value encoded data where the type signals to the RBridge how to interpret the value, and the length tells the RBridge the length of the value in bytes. The type and length are both 16 bit fields. A length of zero indicates the value is a UTF-8 string with a NULL (\0) terminating byte. Preceding the list of TLVs is a 16 bit total length field which specifies the total length of all the length fields in octet units. TLVs with an unknown Type MUST be ignored and skipped over. The value field is 1 byte aligned.
Each TLV in the TLV List appears on the wire encoded as follows:

```
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
```

<table>
<thead>
<tr>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

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

The type values are:

- 0: Next Hop Information, See Section 8.1
- 1: Previous Hop Information, See Section 8.2
- 2: Outgoing Port ID, See Section 8.4
- 3: Incoming Port ID, See Section 8.3
- 4: Outgoing Port MTU, See Section 8.5
- 5: IS-IS System ID, See Section 8.6
For traceroutes targeting known unicast destinations, hop-count errors, and MTU errors, this TLV MUST be a 16-bit nickname and 48-bit system ID of the next hop RBridge the frame is being or would have been sent to. If the next hop RBridge has not reserved a nickname the nickname field must be 0x0000. If the RBridge transmitting the TLV is the egress RBridge this TLV is not included in the frame. For pings, this field MUST be set to all zeros on transmission and ignored on reception. If an RBridge has multiple nicknames it SHOULD use the numerically largest nickname in the Next Hop Information TLV.
8.2. Previous Hop Information

For traceroutes targeting known unicast destinations, hop-count errors, and MTU errors, this TLV MUST be a 16-bit nickname and 48-bit system ID of the previous hop RBridge the frame being responded to was forwarded from. If an RBridge has multiple nicknames it SHOULD use the numerically largest nickname in the Previous Hop Information TLV.

8.3. Incoming Port ID

This TLV MUST be set to the Port ID found in ‘The Special VLANS and Flags sub-TLV’ for the port the request being replied to was received on [RFC6326].
8.4. Outgoing Port ID

```
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|11|12|13|14|15|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                  Type = 0x02                  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                 Length = 0x02                 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                 Outgoing Port ID               |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Outgoing Port ID Format

Figure 11

This TLV MUST be set to the Port ID found in 'The Special VLANs and Flags sub-TLV' for the port the frame is being forwarded on to (or would have been for an echo request/hop-count error) [RFC6326]. If the request was consumed by the replying RBridge, the port ID MUST be 0xFFFF.

8.5. Outgoing Port MTU

```
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|11|12|13|14|15|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                  Type = 0x03                  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                 Length = 0x02                 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                 Outgoing Port MTU              |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Outgoing Port MTU Format

Figure 12

This TLV MUST be the MTU of the outgoing port specified in the outgoing port ID TLV.
8.6. IS-IS System ID

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>+---------------+-------------------+-------------------+-------------------+--------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type = 0x04</td>
<td>Length = 0x06</td>
<td>IS-IS System ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 13

This TLV MUST include the IS-IS System ID of the system generating the message. This TLV MAY be included in all/any messages.

9. Acknowledgments

Many people have contributed to this work, including the following, in alphabetic order: Sam Aldrin, Dinesh Dutt, Donald E. Eastlake 3rd, Anoop Ghanwani, Meenakshi Kaushik, Jeff Laird, Thomas Narten, Santosh Rajagopalan, Marc Sklar, and Li Yizhou.

10. IANA Considerations

IANA is request to create a new subregistry within the TRILL Parameter registry for "TRILL OAM Message Error Sub-Message Error Specifiers". This subregistry that is initially populated as specified in Section 7.2.1. Additional values are allocated by IETF Review [RFC5226].

IANA is requested for a new subregistry within the TRILL Parameter registry for "TRILL Error Reporting Protocol TLV Types" with initial values as listed in Section 5.3. Additional values are allocated by IETF Review [RFC5226].

This draft also requires action to reserve the TRILL RBridge Channel protocol codes. IANA is requested to allocate the TRILL RBridge Channel protocol codes for as listed in Section 5.1.
11. Security Considerations

The nature of the OAM Messages can lead to security concerns. By providing information about the topology and status of a network, attackers can gain greater knowledge of a network in order to exploit the network. Passive attacks such as reading frames with an OAM message could be used to gain such knowledge or active attacks where an attacker mimics an RBridge can be used to probe the network. Authentication, data integrity, protection against replay attacks, and confidentiality for TRILL OAM frames may be provided using a to-be-specified TRILL Security Option. Using such a security option would mitigate both the passive and active attacks.

For instance, data origin authentication could be provided in the future using a security options in the TRILL Header by verifying a hash using shared keys or a mechanism like SEND with CGA. To prevent replay attacks rate limiting, sequence numbers as well as some nonce based mechanism could be provided. Confidentiality for TRILL OAM frames could be provided based on some future security option extension which encrypts TRILL frames.

In a network where one does not wish to configure a security option, the threat of attackers is still present. For this reason, generation of any TRILL OAM Message frames is optional and SHOULD be configurable by an operator on a per RBridge basis. An RBridge MAY have this configurable on a per port basis. For instance, an operator SHOULD be able to disable hop-count traceroute reply messages or error notification message generation per port.

Another security threat is denial of service through use of OAM messages. For this reason, RBridges MUST rate limit the generation of OAM message frames. For multi-destination frames, the frames MAY be discarded silently to prevent any denial of service attacks in case of an error packet such as an 'options not recognized' error notification.

12. References

12.1. Normative References


12.2. Informative References


Appendix A. Implementation Considerations

This appendix contains a few considerations implementors should take note of when creating their user interface as well as some examples of what occurs when a traceroute or ping are executed. These provide a sample user interface one can use as the basis for their user interface.

First, an RBridge SHOULD maintain counters for each type of error generated. There SHOULD be a way for users to view these counters.

Some of the set of default field values for self originated frames are presented in Section 6.2. RBridges SHOULD be configurable to change these values to assign the TRILL data frame to a flow.

A.1. Hop Count Traceroute Example

Figure 14 contains a campus with three RBridges. Consider a hop-count traceroute from RB0 to RB2.

```
+-----+  +-------+   +-------+   +-------+  +-----+
| ESa +--+  RB0  +---+  RB1  +---+  RB2  +--+ ESb |
+-----+  |ingress|   |transit|   |egress |  +-----+
          +-------+   +-------+   +-------+
```

Time       RB0         RB1         RB2
.         (1)------>   |           |
.          | <------- (2)          |
.         (3)------> (3) -------> |
.          | <------- (4) <-------(4)

Hop Count Traceroute Example Topology

Figure 14
In this diagram RB0 transmits frame (1) destined to RB2. This frame contains the echo request message and a hop-count of 0. When RB1 receives this frame it drops it and transmits a hop-count-exceeded message, (2), to RB0. RB0 then transmits a frame, (3), with a hop-count of 1. RB1 decrements this hop-count by 1 to 0 and forwards it to RB2. RB2 drops frame (3) and transmits a Hop Count Zero error notification, (4), to RB0. The traceroute is now complete.

Below are some select fields for the frames:

<table>
<thead>
<tr>
<th>Frame #</th>
<th>Ingress RBridge</th>
<th>Egress RBridge</th>
<th>TRILL OAM Protocol</th>
<th>Sequence Number</th>
<th>Hop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) RB0</td>
<td>RB0</td>
<td>RB2</td>
<td>Echo Request</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(2) RB1</td>
<td>RB1</td>
<td>RB0</td>
<td>Hop Count Zero error notification</td>
<td>1</td>
<td>Default</td>
</tr>
<tr>
<td>(3) @ RB1</td>
<td>RB0</td>
<td>RB2</td>
<td>Echo Request</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>(3) @ RB2</td>
<td>RB0</td>
<td>RB2</td>
<td>Echo Request</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>(4) @ RB1</td>
<td>RB2</td>
<td>RB0</td>
<td>Hop Count Zero error notification</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>(4) @ RB0</td>
<td>RB2</td>
<td>RB0</td>
<td>Hop Count Zero error notification</td>
<td>2</td>
<td>Default</td>
</tr>
</tbody>
</table>

Table 1: Hop Count Traceroute Example Frames

For example, if the nicknames for RB0, RB1, and RB2 are 0x1111, 0x2222, and 0x3333 respectively, the console output from such a trace might be:
Hop Count Tracing

<table>
<thead>
<tr>
<th>RBridge</th>
<th>Incoming Port Id</th>
<th>Outgoing Port Id</th>
<th>RBridge Nexthop Nickname</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1111</td>
<td>Ingress</td>
<td>0x0001</td>
<td>0x2222</td>
</tr>
<tr>
<td>0x2222</td>
<td>0x0000</td>
<td>0x0001</td>
<td>0x3333</td>
</tr>
<tr>
<td>0x3333</td>
<td>0x0000</td>
<td>Egress</td>
<td>0x0000</td>
</tr>
</tbody>
</table>

Table 2: Hop Count Traceroute Example Output

In this example, the first line of output is generated from local information, no hop-count frames are sent to generate it.

A.2. Ping Example

Figure 15 contains a campus with three RBridges. Consider a ping from RB0 to RB2.

```
+-----+  +-------+   +-------+   +-------+  +-----+
| ESa +--+  RB0  +---+  RB1  +---+  RB2  +--+ ESb |
+-----+  |ingress|   |transit|   |egress |  +-----+

Time   RB0   RB1   RB2
.       (1)----> (1)----> |
.       |<-------- (2) <--------(2)

Ping Example Topology

Figure 15
```

In this diagram RB0 transmits frame (1) destined to RB2. This frame contains the echo request message. When RB1 receives this frame it forwards it to RB2. When RB2 receives this frame it forwards it to RB0.

Below are some select fields for the frames:
Table 3: Ping Example Frames

For example, if the nicknames for RB0, RB1, and RB2 are 0x1111, 0x2222, and 0x3333 respectively, the console output from such a ping might be:

Pinging
-----------------------------------
... from 0x1111 to 0x3333... 0x3333 is alive
... from 0x1111 to 0x3333... 0x3333 is alive
... from 0x1111 to 0x3333... 0x3333 is alive

Table 4: Ping Example Output

In this example, the ping was repeated three times with the sequence number (not shown) being changed each time.

Appendix B. Revision History

RFC Editor: Please delete this appendix before publication.

B.1. Changes from -01 to -02

Moved the values table to the message format section and converted from table to list.

Added previous hop information TLV.

Removed error codes that were not needed.

Added path sharing traceroute with 'real' data being sent.

Added mention of BFD draft.

Made most TLVs optional to allow hardware/fast path implementations where this information might not be available.
Changed Next Hop Nickname TLV into Next Hop Information TLV since
next hop might not always reserve a nickname. The new TLV
includes the next hop system id.

Numerous minor typo corrections and wording clarifications.

B.2. Changes from -00 to -01

Broke down the table "frame field values" into two tables,
"response frame field values" and "self initiated frame field
values".

Reorganized the document to move user interface related items to
the appendix and switched the order of ping/traceroute.

Numerous minor typo corrections and wording clarifications.

Authors’ Addresses

David Michael Bond
International Business Machines
2051 Mission College Blvd.
Santa Clara, CA  95054
US

Phone: +1-603-339-7575
EMail: mokon@mokon.net
URI:   http://mokon.net

Vishwas Manral
Hewlett-Packard Co.
19111 Pruneridge Ave.
Cupertino, CA  95014
US

Phone: +1-408-447-0000
EMail: vishwas.manral@hp.com
Pro-active connectivity monitoring for TRILL
draft-rohit-trill-proactive-oam-00.txt

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on August 24, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Abstract

Pro-active fault monitoring for TRILL monitors all the paths between any two given R Bridges in the network. Number of paths to be monitored can be of exponential order based on the distance between two R Bridges. In this document novel fault monitoring mechanism based on a distributed approach is presented.

Table of Contents

1. Introduction...................................................2
   1.1. Contributors..............................................3
2. Conventions used in this document..............................3
3. Motivation.....................................................3
4. Solution overview..............................................6
   4.1. Details for monitoring paths upto 2nd hop Rbridge........8
5. Frame formats..................................................9
   5.1.1. Pro-active fault monitoring request..................9
   5.1.2. Pro-active Payload discovery request................10
   5.1.3. Pro-active Payload discovery response...............12
   5.1.4. Pro-active fault notification.......................13
6. Formal Syntax.................................................16
7. Security Considerations.......................................16
8. IANA Considerations...........................................16
9. Conclusions...................................................16
10. References...................................................16
   10.1. Normative References....................................17
   10.2. Informative References..................................17
11. Acknowledgments..............................................17
Appendix A. Sample report........................................19
   A.1. Summary Report per monitor...............................19
   A.2. Detail Report............................................19
      A.2.1. <H2>................................................20
         A.2.1.1. <H3>...........................................20
            A.2.1.1.1. <H4>......................................20
               A.2.1.1.1.1. <H5>.................................20
         A.2.1.1.1. <H4>......................................20
         A.2.1.1.1.1. <H5>......................................20
   A.3. C type usage in messages................................20

1. Introduction

Pro-active fault monitoring is necessary for all OAM solutions. It gives network service providers confidence about the health of their network. Whenever network service is provided to customers with SLA
(Service Level Agreement), it becomes even more important to monitor the network pro-actively. In traditional Layer-2 networks (CE) pro-active fault monitoring is done based on VLANs. Since spanning-tree ensures that there is a single path between any two nodes, it is straightforward mechanism to monitor path between 2 given RBridges and given VLAN.

TRILL Base Protocol Specification [RFC6325] provides a method for forwarding Layer 2 data frames over multiple active links. There can be number of ECMPs (Equal Cost Multiple Paths) between any two given TRILL RBridges. As the number of hops between given two RBridges increases, number of ECMPs increases exponentially. Pro-active monitoring in this case needs to monitor all the ECMPs between two given RBridges.

TRILL OAM draft [TRILLOAM] proposes OAM suite for TRILL. This draft is for adding pro-active functionality to the OAM suite. It extends C-types defined in TRILL OAM draft, for pro-active monitoring.

1.1. Contributors

Chandan Mishra has contributed with ideas and comments for devising the solution presented in this document.

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].

3. Motivation

As we discussed in the introduction, number of paths to be tested increases exponentially as the number of hops between ingress and egress RBridge increases. Identifying the header parameters (mac/IP/L4 addresses) to exercise each unique path is a hard problem and needs information about hashing functions from each intermediate RBridge.

Sending test packets, with random header parameters, expecting that will exercise different ECMPs is one option. But in this case number of packets that need to be sent can be even higher than total number of ECMPs.

In this document we take a different approach to address this problem.
Testing end-to-end connectivity means, testing connectivity of all links along the path as well as exercising switching on all intermediate RBridges. Instead of doing it end to end, same can be done after splitting it into multiple short paths, such that, paths overlap to cover complete end-to-end connectivity. If each such short path is limited only to two hops, it brings down number of packets to be sent from exponential order of number of hops \((k^n)\) to order of \(k*n\), where \(k\) is assumed to be number of ECMPs for each hop for simplification of calculation and \(n\) is number of hops between the Ingress and Egress RBridges.

Consider following Figure 1. In the figure, Rbridges are numbered from 1 to 10. The problem is to monitor end-to-end connectivity between Rbridge 1 and 10.

![Figure 1 Example network.](image)

In above Figure 1, RBridges 2 and 3 are connected to RBridges 4 and 5, RBridges 4 and 5 are connected to RBridges 6 and 7. Rbridges 6 and 7 are connected to RBridges 8 and 9.

Total number of ECMPs between RBridge 1 and RBridge 10 is -
\[
2 \times 2 \times 2 \times 2 \times 2 = 32
\]

Hence Rbridge 1 will need to send 32 packets to test all ECMPs to Rbridge 10, assuming Rbridge 1 knows payloads required to exercise all these paths.

Above network can be split into four overlapping sections as shown in Figure 2 and Figure 3. If we test all paths between Ingress and Egress Rbridges of each section then all paths between 1 and 10 will be tested.
In Figure 2, for the section 1, the number of paths between RBridges 1 and 4 is 2 and number of paths between RBridges 1 and 5 is 2. Hence total number of paths to be tested for sub-network1 is 4.
Similarly, number of paths between RBridges 2 and 6 is 2, between RBridges 2 and 7 is 2. Number of paths between RBridges 3 and 6 is 2 and between RBridges 3 and 7 is 2.

Hence total number of paths to be tested is 8.

Similarly from Figure 3, total number paths to be tested for section 3 is 8 and for section 4 is 4.

Note that, in the above example network, maximum number of paths to be tested by any given Rbridge is limited to 8. Hence load of monitoring network is now distributed. Also total number of paths tested is 4+8+8+4=24.

Note that if Rbridge 1 was to do testing for all paths, number of paths to be tested would be 32. As the complexity of the network increases and number of paths between Ingress and Egress Rbridges increases, the mechanism proposed here will yield even more benefits.

4. Solution overview

Here we present high level overview of the solution. More details are discussed in the subsequent sections.

Pro-active fault monitoring is initiated by the user. As part of the request, user identifies a VLAN and 2 RBridges - Ingress and Egress Rbridges. All Equal Cost Paths ECMPs on this vlan and between these two RBridges need to be monitored. User provides total time interval for monitoring session as the part of the request.

Here are the high-level steps of the mechanism

1. Ingress Rbridge starts connectivity tests for paths upto its 2nd hop Rbridge(s)(on the path to egress RBridge).

2. If 2nd hop Rbridge is egress Rbridge, it stops the test.

3. Else it requests its 1st hop Rbridge(s) (on the path to egress), to initiate the tests, starting with step1.

Once the request is distributed, whenever a fault is detected, it is indicated to the Originator Rbridge using a fault notification message which includes fault details.

Consider Figure 4 as example network. Let us assume user requests proactive fault monitoring between ingress RBridge RB1 and egress
RBridge RB5. P1 to P5 are the Egress ports along the ECMPs between RB1 and RB5.

As per step1, RB1 tests all paths upto its 2nd hop Rbridge on the path along RB5.

For that, RB1 sends 'payload request' message to its 1st hop Rbridges on the path along RB5. RB1 looks up its local forwarding table, and finds that p1 is Egress port for path towards RB5. It then sends 'payload request' with TTL=1 on p1. RB2, will reply back with 2 payloads say PL1 and PL2, for taking path along ports p3 and p2, respectively.

RB1 now sends two packets with payloads PL1 and PL2, and TTL=2. When RB1 receives 'hop count expiry' message for both, it confirms that paths up to its 2nd hop Rbridge(s) are fault-free (i.e. paths between RB1 and RB3, as well as between RB1 and RB4 are fault-free).

As per step3, RB1 also forwards the 'pro-active fault monitoring request' message defined in section 5.1.1, to monitor connectivity, to its 1st hop Rbridges along the path. It does so by sending request with TTL=1, on port p1.
RB2 on receiving this request, will repeat the step1. It will send 'payload request' with TTL=1, on port p2 and p3. For each request, it will get one payload, say PL3 for request sent on p2 and PL4 for request sent on p2. It will test paths up to its 2nd hop Rbridges, by sending a packet with payload PL3 on port p2 and a packet with payload PL4 on port p3 and TTL=2.

As RB2 will receive 'hop count expiry' message from RB5, it will not forward the requests for monitoring paths till RB5, to its 1st hop Rbridges.

4.1. Details for monitoring paths upto 2nd hop Rbridge

For a given egress TRILL RBridge, local TRILL routing table can provide information about different next hop Rbridges/Egress ports to exercise the ECMPs.

We propose to send 'Payload Discovery request' message on each of these ports, with TTL=1. 'Payload Discovery request' (section 5.1.1.) message carries information about egress TRILL RBridge (RB5) in the original request made by the user.

Based on the egress RBridge (RB5), each 1st hop RBridge looks up its TRILL forwarding tables, and for each equal cost multi path towards egress RBridge (RB5) identifies a unique inner destination MAC addresses, that will exercise the ECMPs towards egress RBridge-id. These MAC addresses will be sent back in a 'Payload Discovery response packet'.

The source mac address is not used for payload generation, as it might be learnt by other RBridge, if the packets are originated by TRILL Edge RBridges. Well known source mac address is used, so that it will not be used by any real data packets. Ethtype is fixed to TRILL OAM Diagnostic ethtype to restrict these frames from leaving TRILL network (refer section 6.2, from [TRILLOAM]). VLAN is specified by the user as a part of the request. For payload generation, nickname of the requester RBridge, provided in 'payload generation request'(section 5.1.2), is used as ingress RBridge nickname. Egress RBridge nickname provided in the request is used as Egress RBridge nickname for payload generation.

The current TRILL RBridge, receives list of destination mac addresses on each port on ECMP. It constructs 'loopback message' (TRILLOAM) message with TTL=2 and these mac addresses as inner destination mac addresses and sends these on ports on which corresponding mac address was received.
Each packet sent, will now test the switching at next hop and test all links on the path taken by the packet. The inband or out of band ICMP ‘hop count expiry response’ (TRILLOAM), will confirm that both are fault-free. When all such responses are received, it will confirm that all paths towards egress Rbridge are error free, till 2nd hop. If there is a fault, fault details are sent to the originator Rbridge. If current Rbridge itself is the originator Rbridge, it saves the fault information.

Payload generation request is sent periodically based on the ‘Payload generation request time interval’ specified in the user request. Test packets are also sent at an ‘test time interval’ provided by the user. Finally this whole monitoring process is continued till the ‘Monitor time interval’, also specified by the user. ‘Pro-active fault monitoring request’ defined in next section is used for forwarding the request to 1st hop Rbridges.

5. Frame formats

5.1.1. Pro-active fault monitoring request

Pro-active fault monitoring request includes C-type 44 which provides Egress Rbridge ID and originator Rbridge ID. It also provides information about timers required in fault monitoring. C-type 4 (interested vlan) is included in the request to indicate monitored vlan, if the request packet is not using the same vlan. Source Mac address and ethtype are fixed to the values used in the request packet. Pro active fault monitoring message is represented by TRILL OAM message code 26.

![Frame Format](image_url)

Figure 5 Pro-active fault monitoring request details (C-type 44)
Egress nickname (2 octets): nickname of the Egress/egress Rbridge.

Originator nickname (2 octets): nickname of the originator Rbridge.

Ingress nickname (2 octets): nickname of the ingress Rbridge.

S (1 bit), start/stop request: if set to 1, specifies request to start monitoring, if set to 0, specifies request to stop monitoring.

G (1bit); Global stop, when set to 1, stops all pro-active monitoring requests on this Rbridge requested by same Originator RBridge, irrespective of other information in the C-type. Set to 1 only for debugging.

Reserved (14 bits):

MonitorId (16bits): Identifier for the current session. It is generated by Originator Rbridge such that it is unique locally, and propagated while forwarding request to next hops. MonitorId, combined with Originator Rbridge ID, forms unique identifier for fault monitoring session.

Monitor interval (4octets): total interval of fault monitoring session in seconds. 0 is a special value, indicating it needs to run till request to stop comes.

Payload Generation Interval(2 octets): interval for refreshing payload parameters by sending payload generation request in seconds.

Test interval (2 octets): interval for sending test packets with TTL=2, for testing paths till 2nd hop in seconds.

5.1.2. Pro-active Payload discovery request

C-type ‘Payload discovery request for pro-active monitoring’ is different from Payload Discovery request defined in section 8.2 in [TRILLOAM]. This C-type by definition allows use of any Destination Mac address for payload generation. It also expects that response will include payloads for exercising all available ECMPs. Along with this new type, interested vlan (ctype 4) is also specified, if packet is not using same vlan. Source Mac address and ethtype are fixed to the values used in the request packet. Payload discovery message is represented by TRILL OAM message code (22).
Figure 6 Pro-active Payload Discovery request (C-type 45)

Egress nickname (2 octets): nickname of the Egress Rbridge provided by user (used as Egress Rbridge nickname for payload generation).

Originator nickname (2 octets): nickname of the originator Rbridge.

Ingress nickname (2 octets): nickname of the ingress Rbridge.

Requester nickname (2 octets): nickname of the Rbridge sending this request (Used as Ingress nickname for payload generation).
5.1.3. Pro-active Payload discovery response

‘Payload generation response for Proactive monitoring’ specifies one or more Destination MAC addresses, one for each ECMP. Its uses new C-type 46 which lists down destination mac addresses (DMAC1, DMAC2..DMACn where n is number of ECMPs). TRILL OAM code is set to payload generation response (23).

<table>
<thead>
<tr>
<th>egress nickname</th>
<th>S</th>
<th>ECMP count</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMAC1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMAC1</td>
<td></td>
<td>next hop nickname</td>
<td>R</td>
</tr>
<tr>
<td>ifindex</td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>Port</td>
<td></td>
<td>Slot</td>
<td>a</td>
</tr>
<tr>
<td>State</td>
<td></td>
<td>Speed</td>
<td></td>
</tr>
</tbody>
</table>

DMAC and Next hop path information (from ifindex to speed) repeated for number for all ECMPs.

Figure 7 Pro-active Payload Discovery Response (C-type 46)

Egress nickname (2 octets): nickname of the Egress Rbridge.

S (3 bits): indicates the status

0. Success
1. ECMP data not found
2. System overloaded try later
3. -7 Reserved MUST not be used

ECMP count (8bit) - specifies number of ECMPs

DMAC1-DMACn - Destination mac addresses, (number of MAC addresses is equal to number if ECMP count).

Next hop nickname (2 octets): nickname of the next hop Rbridge, to which packet will be forwarded if Destination mac given with this field is used.
Ifindex (4 octets) : unsigned integer of local significance
Slot  (2 octets) : Slot number
Port   (2 octets) : Port number
Speed  (2 octets) : Speed in 100Mbps. Zero (0) indicates port speeds less than 100Mbps.
State  (2 octets) : Represent the state of the port.
0: Down - no errors
1: Disable
2: Forwarding-no errors
3: Down - errors
4: Forwarding - errors
5: Forwarding - oversubscribed
6: Link monitoring disable
All other values reserved.

Total number of DMAC and next hop path information entries MUST be equal to ECMP count.

5.1.4. Pro-active fault notification

Fault details are sent to the originator Rbridge provided in the 'proactive monitoring request' by including C-type 47 (downstream identification for pro-active monitoring).

C type 47 gives information about interface on which connectivity test failed, i.e, hop count expiry message was not received.

If 'payload discovery generation response', had succeeded, one more copy of C_type 47 is included in the pro-active fault notification. The fields in this entry, are copied from the 'payload discovery generation response'. If connectivity was being tested using DMAC3 (DMAC provided in the 3rd entry in payload discovery response), the
details of the interface provided with the DMAC3 will be used in this instance of C type 47.

The notification can be sent either inband or out of band. TRILL OAM code is set to parameter problem notification (5).

```
0                                  31
+----------------------+-------------------+
| S |     Reserved1    | responder-nickname|
+----------------------+-------------------+
| Local nickname       |  next hop nickname|
+----------------------+-------------------+
|              ifindex                     |
+----------------------+-------------------+
|      Port             |      Slot         |
+----------------------+-------------------+
|     State           |      Speed        |
+----------------------+-------------------+
```

Figure 8 downstream identification for pro-active monitoring (C-type 47)

Responder nickname (16 bits): TRILL 16 bit nickname of responder RBridge [RFC3804]

S (3 bits): 'Payload discover generation response' status from section 5.1.3. If Status is not Success, remaining fields below can be set to 0.

0. Success
1. ECMP data not found
2. System overloaded try later
3. Not available
4. 4-7 Reserved MUST not be used

ECMP count (8 bits): Copied from for 'payload generation response', if Status was 'Success'.

Reserved1 (13 bits): Reserved, set to zero on transmission and ignored on receipt.

Next-hop neighbor information:
Local Nickname (16 bits): TRILL 16 bit nickname of the local RBridge [RFCtrill]

Next hop Nickname (16 bits): TRILL 16 bit nickname of the next hop RBridge[RFCtrill]

Slot (2 octets) : Slot number

Port (2 octets) : Port number

Speed (2 octets) : Speed in 100Mbps. Zero (0) indicates port speeds less than 100Mbps.

State (2 octets) : Represent the state of the port.

0: Down - no errors
1: Disable
2: Forwarding-no errors
3: Down - errors
4: Forwarding - errors
5: Forwarding - oversubscribed
6: Link monitoring disable

All other values reserved.

1 instance of C-type 47 gives information about interface connecting local RBridge and 1st hop RBrigdes.

If ‘payload generation response’ status (section 5.1.3), was SUCCESS, one more instance of C-type 47 is also included as part of ‘pro-active fault notification.

2nd instance of C-type 47 gives information about interface connecting 1st hop RBridge and 2nd hop RBrigdes, that ‘loopback message’ (TRILLOAM) message would have encountered, if connectivity test was successful (i.e if hop count expiry message was received from 2nd hop Rbrige).

If ‘loopback message’ message (TRILLOAM) used Destination Mac address provided in the nth entry of ‘payload generation response’ (section 5.1.3), 2nd instance of Ctype 47 will include interface
information provided in that particular entry. In this instance of C-type 47, Status is set to 3 (not available).

6. Formal Syntax

INFO (REMOVE): Include this section only if needed. Commonly used grammar is BNF grammar defined in RFC-2234. Suggested wording.

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in RFC-2234 [RFC2234].

<Define your formal syntax here.>

7. Security Considerations

INFO (REMOVE): Every draft MUST have a Security Considerations section.

Security consideration for pro-active monitoring are similar to TRILL OAM draft [TRILLOAM]. Request/response packet can not go out of the TRILL cloud, as TRILL OAM ethtype packets are dropped at the Edge Rbridge. Since Pro-active monitoring request can be issued only at a TRILL Rbridge, consideration is needed only for ensuring packet with TRILL OAM ethtype and c-type 43 is dropped at Ingress Rbridge.

8. IANA Considerations

INFO (REMOVE): Every draft MUST have an IANA Considerations section, although it may be removed prior to publication by the RFC Editor if null.

<Add any IANA considerations>

9. Conclusions

<Add any conclusions>

10. References

INFO (REMOVE): Authors can use either the auto-numbered references OR the named references; typically, these would not be mixed in a single document. This template includes both examples for illustration of the two variations. Named references are preferred (e.g., [RFC2119] or [Fab1999]).
10.1. Normative References

INFO (REMOVE): Normative refs are references to standards documents **required** to understand this doc. These are usually Standards-track and BCP RFCs, or external (IEEE, ANSI, etc.) standards, but may include other publications.


10.2. Informative References

INFO (REMOVE): Informative refs are those that are not standards or standards not required to understand this doc. These are usually informative RFCs, internet-drafts (avoid if possible), and other external documents.

[RFC792] Postel, J. "Internet Control Message Proctocol (ICMP)", RFC 792, September, 1981.


11. Acknowledgments

<Add any acknowledgements>
INFO (REMOVE): The author of this template would appreciate if you would keep the following line in your final IDs and RFCs:

This document was prepared using 2-Word-v2.0.template.dot.
Appendix A. Sample report

INFO (REMOVE): Starts on a new page. These are optional.

INFO (REMOVE): Careful with headers in appendices - they won’t renumber when moved in/out levels in outline mode. Only Headers 1-9 do that trick, as used in the body of the RFC!

A.1. Summary Report per monitor

Monitor Vlan:

Monitor paths to RbridgeID:

Monitor Id:

Monitoring for time: x days x hours x minutes x seconds

Total Faults reported : x faults

Faulty paths detected (RbridgeId1 to RbridgeId3)

(RbridgeId1/Interface)-> (RbridgeId2/Interface)->RbridgeId3

S1/eth3/2           S2/eth4/5      S3
S4/eth5/2           S5/eth4/5      S3

A.2. Detail Report

Fault detection log:

2012 Jan 31 13:50:34 Fault at (S1/eth3/2,S2) "ECMP information not found" for monitor to S7, vlan 2, MonitorId 10.

2012 Jan 31 13:51:24 Fault at (S4/eth5/2,S5/eth4/5,S3) "Packet flow test failed" for monitor to S8, vlan 1, MonitorId 3.

2012 Jan 31 13:52:24 Fault at (S4/eth5/2,S5/eth4/5,S3)"Packet flow test failed" for monitor to S8, vlan 1, MonitorId 3.
A.2.1.  <H2>

<Text>

A.2.1.1.  <H3>

<Text>

A.2.1.1.1.  <H4>

<Text>

A.2.1.1.1.1.  <H5>

<Text>

A.3.  C type usage is messages

<table>
<thead>
<tr>
<th>Message</th>
<th>Mandatory parameters</th>
<th>Optional parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proactive fault monitoring request</td>
<td>Version(1)</td>
<td>Interested vlan(4)</td>
</tr>
<tr>
<td></td>
<td>Pro-active fault</td>
<td></td>
</tr>
<tr>
<td></td>
<td>monitoring request</td>
<td></td>
</tr>
<tr>
<td></td>
<td>details(44)</td>
<td></td>
</tr>
<tr>
<td>Proactive fault discovery request</td>
<td>Version(1)</td>
<td>Interested vlan(4)</td>
</tr>
<tr>
<td></td>
<td>Pro-active fault</td>
<td></td>
</tr>
<tr>
<td></td>
<td>discovery request</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(45)</td>
<td></td>
</tr>
<tr>
<td>Proactive fault discovery response</td>
<td>Version(1)</td>
<td>Pro-active fault</td>
</tr>
<tr>
<td></td>
<td>notification(46)</td>
<td>notification(47)</td>
</tr>
<tr>
<td></td>
<td>(2nd instance)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 9 Optional and mandatory C-types.

New TRILL OAM code 26: Pro-active fault monitoring request
Authors’ Addresses

Rohit Watve  
CISCO Systems  
375 East Tasman Drive,  
San Jose, CA 95134  
Phone: 408-424-2091  
Email: rwatve@cisco.com

Tissa Senevirathne  
CISCO Systems  
375 East Tasman Drive,  
San Jose, CA 95134  
Phone: 408-853-2291  
Email: tsenevir@cisco.com

Gayatri Ramachandran  
CISCO Systems  
375 East Tasman Drive,  
San Jose, CA 95134  
Phone: 408-424-0828  
Email: garamach@cisco.com
Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on April 28, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document.

Sajassi, et. al.
Abstract
This document discusses how Ethernet Provider Backbone Bridging [802.1ah] can be combined with E-VPN in order to reduce the number of BGP MAC advertisement routes by aggregating Customer/Client MAC (C-MAC) addresses via Provider Backbone MAC address (B-MAC), provide client MAC address mobility using C-MAC aggregation and B-MAC sub-netting, confine the scope of C-MAC learning to only active flows, offer per site policies and avoid C-MAC address flushing on topology changes. The combined solution is referred to as PBB-EVPN.

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

Table of Contents
1. Introduction.................................................... 3
2. Contributors.................................................... 4
3. Terminology..................................................... 4
4. Requirements.................................................... 4
4.1. MAC Advertisement Route Scalability........................... 4
4.2. C-MAC Mobility with MAC Sub-netting........................... 5
4.3. C-MAC Address Learning and Confinement........................ 5
4.4. Interworking with TRILL and 802.1aq Access Networks with C-MAC
     Address Transparency........................................... 5
4.5. Per Site Policy Support....................................... 6
4.6. Avoiding C-MAC Address Flushing............................... 6
5. Solution Overview............................................... 6
6. BGP Encoding.................................................... 7
6.1. BGP MAC Advertisement Route.................................. 7
6.2. Ethernet Auto-Discovery Route................................ 8
6.3. Per VPN Route Targets.......................................... 8
6.4. MAC Mobility Extended Community.............................. 8

Sajassi, et al.
1. Introduction

[E-VPN] introduces a solution for multipoint L2VPN services with advanced multi-homing capabilities using BGP for distributing customer/client MAC address reachability information over the core MPLS/IP network. [802.1ah] defines an architecture for Ethernet Provider Backbone Bridging (PBB), where MAC tunneling is employed to improve service instance and MAC address scalability in Ethernet networks and in VPLS networks [PBB-VPLS].
In this document, we discuss how PBB can be combined with E-VPN in order to reduce the number of BGP MAC advertisement routes by aggregating Customer/Client MAC (C-MAC) addresses via Provider Backbone MAC address (B-MAC), provide client MAC address mobility using C-MAC aggregation and B-MAC sub-netting, confine the scope of C-MAC learning to only active flows, offer per site policies and avoid C-MAC address flushing on topology changes. The combined solution is referred to as PBB-EVPN.

2. Contributors

In addition to the authors listed above, the following individuals also contributed to this document.

Keyur Patel
Cisco

3. Terminology

BEB: Backbone Edge Bridge
B-MAC: Backbone MAC Address
CE: Customer Edge
C-MAC: Customer/Client MAC Address
DHD: Dual-homed Device
DHN: Dual-homed Network
LACP: Link Aggregation Control Protocol
LSM: Label Switched Multicast
MDT: Multicast Delivery Tree
MES: MPLS Edge Switch
MP2MP: Multipoint to Multipoint
P2MP: Point to Multipoint
P2P: Point to Point
PoA: Point of Attachment
PW: Pseudowire
E-VPN: Ethernet VPN

4. Requirements

The requirements for PBB-EVPN include all the requirements for E-VPN that were described in [EVPN-REQ], in addition to the following:

4.1. MAC Advertisement Route Scalability

In typical operation, an [E-VPN] MES sends a BGP MAC Advertisement Route per customer/client MAC (C-MAC) address. In certain applications, this poses scalability challenges, as is the case in virtualized data center environments where the number of virtual machines (VMs), and hence the number of C-MAC addresses, can be in the millions. In such scenarios, it is required to reduce the number of BGP MAC Advertisement routes by relying on a MAC ‘summarization’
scheme, as is provided by PBB. Note that the MAC sub-netting
capability already built into E-VPN is not sufficient in those
environments, as will be discussed next.

4.2. C-MAC Mobility with MAC Sub-netting

Certain applications, such as virtual machine mobility, require
support for fast C-MAC address mobility. For these applications, it
is not possible to use MAC address sub-netting in E-VPN, i.e.
advertise reachability to a MAC address prefix. Rather, the exact
virtual machine MAC address needs to be transmitted in BGP MAC
Advertisement route. Otherwise, traffic would be forwarded to the
wrong segment when a virtual machine moves from one Ethernet segment
to another. This hinders the scalability benefits of sub-netting.

It is required to support C-MAC address mobility, while retaining
the scalability benefits of MAC sub-netting. This can be achieved by
leveraging PBB technology, which defines a Backbone MAC (B-MAC)
address space that is independent of the C-MAC address space, and
aggregate C-MAC addresses via a B-MAC address and then apply sub-
netting to B-MAC addresses.

4.3. C-MAC Address Learning and Confinement

In E-VPN, all the MES nodes participating in the same E-VPN instance
are exposed to all the C-MAC addresses learnt by any one of these
MES nodes because a C-MAC learned by one of the MES nodes is
advertise in BGP to other MES nodes in that E-VPN instance. This is
the case even if some of the MES nodes for that E-VPN instance are
not involved in forwarding traffic to, or from, these C-MAC
addresses. Even if an implementation does not install hardware
forwarding entries for C-MAC addresses that are not part of active
traffic flows on that MES, the device memory is still consumed by
keeping record of the C-MAC addresses in the routing table (RIB). In
network applications with millions of C-MAC addresses, this
introduces a non-trivial waste of MES resources. As such, it is
required to confine the scope of visibility of C-MAC addresses only
to those MES nodes that are actively involved in forwarding traffic
to, or from, these addresses.

4.4. Interworking with TRILL and 802.1aq Access Networks with C-MAC
Address Transparency

[TRILL] and [802.1aq] define next generation Ethernet bridging
technologies that offer optimal forwarding using IS-IS control
plane, and C-MAC address transparency via Ethernet tunneling
technologies. When access networks based on TRILL or 802.1aq are
interconnected over an MPLS/IP network, it is required to guarantee
C-MAC address transparency on the hand-off point and the edge (i.e.
MES) of the MPLS network. As such, solutions that require
termination of the access data-plane encapsulation (i.e. TRILL or
802.1aq) at the hand-off to the MPLS network do not meet this transparency requirement, and expose the MPLS edge devices to the MAC address scalability problem.

PBB-EVPN supports seamless interconnect with these next generation Ethernet solutions while guaranteeing C-MAC address transparency on the MES nodes.

4.5. Per Site Policy Support

In many applications, it is required to be able to enforce connectivity policy rules at the granularity of a site (or segment). This includes the ability to control which MES nodes in the network can forward traffic to, or from, a given site. PBB-EVPN is capable of providing this granularity of policy control. In the case where per C-MAC address granularity is required, the EVI can always continue to operate in E-VPN mode.

4.6. Avoiding C-MAC Address Flushing

It is required to avoid C-MAC address flushing upon link, port or node failure for multi-homed devices and networks. This is in order to speed up re-convergence upon failure.

5. Solution Overview

The solution involves incorporating IEEE 802.1ah Backbone Edge Bridge (BEB) functionality on the E-VPN MES nodes similar to PBB-VPLS PEs (PBB-VPLS) where BEB functionality is incorporated in PE nodes. The MES devices would then receive 802.1Q Ethernet frames from their attachment circuits, encapsulate them in the PBB header and forward the frames over the IP/MPLS core. On the egress E-VPN MES, the PBB header is removed following the MPLS disposition, and the original 802.1Q Ethernet frame is delivered to the customer equipment.
The MES nodes perform the following functions:
- Learn customer/client MAC addresses (C-MACs) over the attachment circuits in the data-plane, per normal bridge operation.
- Learn remote C-MAC to B-MAC bindings in the data-plane from traffic ingress from the core per [802.1ah] bridging operation.
- Advertise local B-MAC address reach-ability information in BGP to all other MES nodes in the same set of service instances. Note that every MES has a set of local B-MAC addresses that uniquely identify the device. More on the MES addressing in section 5.
- Build a forwarding table from remote BGP advertisements received associating remote B-MAC addresses with remote MES IP addresses and the associated MPLS label(s).

6.        BGP Encoding

PBB-EVPN leverages the same BGP Routes andAttributes defined in [E-VPN], adapted as follows:

6.1.          BGP MAC Advertisement Route

The E-VPN MAC Advertisement Route is used to distribute B-MAC addresses of the MES nodes instead of the C-MAC addresses of end-stations/hosts. This is because the C-MAC addresses are learnt in the data-plane for traffic arriving from the core. The MAC Advertisement Route is encoded as follows:

- The RD is set to a Type 1 RD RD [RFC4364]. The value field encodes the IP address of the MES (typically, the loopback address) followed by 0. The reason for such encoding is that the RD cannot be that of a single EVI since the same B-MAC address can span across multiple EVIs.
- The MAC address field contains the B-MAC address.
- The Ethernet Tag field is set to 0.

The route is tagged with the set of RTs corresponding to all EVIs associated with the B-MAC address.

All other fields are set as defined in [E-VPN].
6.2. Ethernet Auto-Discovery Route

This route and any of its associated modes is not needed in PBB-EVPN.

6.3. Per VPN Route Targets

PBB-EVPN uses the same set of route targets defined in [E-VPN]. More specifically, the RT associated with a VPN is set to the value of the I-SID associated with the service instance. This eliminates the need for manually configuring the VPN-RT.

6.4. MAC Mobility Extended Community

This extended community is a new transitive extended community. It may be advertised along with MAC Advertisement routes. When used in PBB-EVPN, it indicates that the C-MAC forwarding tables for the I-SIDs associated with the RTs tagging the MAC Advertisement routes must be flushed. This extended community is encoded in 8-bytes as follows:
- Type (1 byte) = Pending IANA assignment.
- Sub-Type (1 byte) = Pending IANA assignment.
- Reserved (2 bytes)
- Counter (4 bytes)

Note that all other BGP messages and/or attributes are used as defined in [E-VPN].

7. Operation

This section discusses the operation of PBB-EVPN, specifically in areas where it differs from [E-VPN].

7.1. MAC Address Distribution over Core

In PBB-EVPN, host MAC addresses (i.e. C-MAC addresses) need not be distributed in BGP. Rather, every MES independently learns the C-MAC addresses in the data-plane via normal bridging operation. Every MES has a set of one or more unicast B-MAC addresses associated with it, and those are the addresses distributed over the core in MAC Advertisement routes. Given that these B-MAC addresses are global within the provider’s network, there’s no need to advertise them on a per service instance basis.

7.2. Device Multi-homing

7.2.1. MES MAC Layer Addressing & Multi-homing

In [802.1ah] every BEB is uniquely identified by one or more B-MAC addresses. These addresses are usually locally administered by the
Service Provider. For PBB-EVPN, the choice of B-MAC address(es) for the MES nodes must be examined carefully as it has implications on the proper operation of multi-homing. In particular, for the scenario where a CE is multi-homed to a number of MES nodes with all-active redundancy and flow-based load-balancing, a given C-MAC address would be reachable via multiple MES nodes concurrently. Given that any given remote MES will bind the C-MAC address to a single B-MAC address, then the various MES nodes connected to the same CE must share the same B-MAC address. Otherwise, the MAC address table of the remote MES nodes will keep flip-flopping between the B-MAC addresses of the various MES devices. For example, consider the network of Figure 1, and assume that MES1 has B-MAC BM1 and MES2 has B-MAC BM2. Also, assume that both links from CE1 to the MES nodes are part of an all-active multi-chassis Ethernet link aggregation group. If BM1 is not equal to BM2, the consequence is that the MAC address table on MES3 will keep oscillating such that the C-MAC address CM of CE1 would flip-flop between BM1 or BM2, depending on the load-balancing decision on CE1 for traffic destined to the core.

Considering that there could be multiple sites (e.g. CEs) that are multi-homed to the same set of MES nodes, then it is required for all the MES devices in a Redundancy Group to have a unique B-MAC address per site. This way, it is possible to achieve fast convergence in the case where a link or port failure impacts the attachment circuit connecting a single site to a given MES.

```
\begin{figure}[h]
\centering
\begin{tikzpicture}
\node at (0,0) {CE1};
\node at (1,0) {M1};
\node at (2,0) {MES1};
\node at (1,-1) {M2};
\node at (2,-1) {MES2};
\node at (3,0) {MES3};
\node at (2,1) {M1};
\node at (2,2) {IP/MPLS};
\node at (2,3) {Network};
\node at (2,4) {MESr};
\draw (0,0) -- (2,0);
\draw (1,0) -- (2,-1);
\draw (2,-1) -- (3,0);
\end{tikzpicture}
\caption{B-MAC Address Assignment}
\end{figure}
```

In the example network shown in Figure 2 above, two sites corresponding to CE1 and CE2 are dual-homed to MES1/MES2 and MES2/MES3, respectively. Assume that BM1 is the B-MAC used for the site corresponding to CE1. Similarly, BM2 is the B-MAC used for the site corresponding to CE2. On MES1, a single B-MAC address (BM1) is required for the site corresponding to CE1. On MES2, two B-MAC addresses (BM1 and BM2) are required, one per site. Whereas on MES3,
a single B-MAC address (BM2) is required for the site corresponding to CE2. All three MES nodes would advertise their respective B-MAC addresses in BGP using the MAC Advertisement routes defined in [EVPN]. The remote MES, MESr, would learn via BGP that BM1 is reachable via MES1 and MES2, whereas BM2 is reachable via both MES2 and MES3. Furthermore, MESr establishes via the normal bridge learning that C-MAC M1 is reachable via BM1, and C-MAC M2 is reachable via BM2. As a result, MESr can load-balance traffic destined to M1 between MES1 and MES2, as well as traffic destined to M2 between both MES2 and MES3. In the case of a failure that causes, for example, CE1 to be isolated from MES1, the latter can withdraw the route it has advertised for BM1. This way, MESr would update its path list for BM1, and will send all traffic destined to M1 only to MES2.

For single-homed sites, it is possible to assign a unique B-MAC address per site, or have all the single-homed sites connected to a given MES share a single B-MAC address. The advantage of the first model over the second model is the ability to avoid C-MAC destination address lookup on the disposition PE (even though source C-MAC learning is still required in the data-plane). Also, by assigning the B-MAC addresses from a contiguous range, it is possible to advertise a single B-MAC subnet for all single-homed sites, thereby rendering the number of MAC advertisement routes required at par with the second model.

In summary, every MES may use a unicast B-MAC address shared by all single-homed CEs or a unicast B-MAC address per single-homed CE, and in addition a unicast B-MAC address per dual-homed CE. In the latter case, the B-MAC address MUST be the same for all MES nodes in a Redundancy Group connected to the same CE.

7.2.1.1. Automating B-MAC Address Assignment

The MES B-MAC address used for single-homed sites can be automatically derived from the hardware (using for e.g. the backplane’s address). However, the B-MAC address used for multi-homed sites must be coordinated among the RG members. To automate the assignment of this latter address, the MES can derive this B-MAC address from the MAC Address portion of the CE’s LACP System Identifier by flipping the ‘Locally Administered’ bit of the CE’s address. This guarantees the uniqueness of the B-MAC address within the network, and ensures that all MES nodes connected to the same multi-homed CE use the same value for the B-MAC address.

Note that with this automatic provisioning of the B-MAC address associated with multi-homed CEs, it is not possible to support the uncommon scenario where a CE has multiple bundles towards the MES nodes, and the service involves hair-pinning traffic from one bundle to another. This is because the split-horizon filtering relies on B-MAC addresses rather than Site-ID Labels (as will be described in
the next section). The operator must explicitly configure the B-MAC address for this fairly uncommon service scenario.

Whenever a B-MAC address is provisioned on the MES, either manually or automatically (as an outcome of CE auto-discovery), the MES MUST transmit an MAC Advertisement Route for the B-MAC address with a downstream assigned MPLS label that uniquely identifies that address on the advertising MES. The route is tagged with the RTs of the associated EVIs as described above.

### 7.2.2. Split Horizon and Designated Forwarder Election

[E-VPN] relies on access split horizon, where the Ethernet Segment Label is used for egress filtering on the attachment circuit in order to prevent forwarding loops. In PBB-EVPN, the B-MAC source address can be used for the same purpose, as it uniquely identifies the originating site of a given frame. As such, Segment Labels are not used in PBB-EVPN, and the egress filtering is done based on the B-MAC source address. It is worth noting here that [802.1ah] defines this B-MAC address based filtering function as part of the I-Component options, hence no new functions are required to support split-horizon beyond what is already defined in [802.1ah]. Given that the Segment label is not used in PBB-EVPN, the MES sets the Label field in the Ethernet Segment Route to 0.

The Designated Forwarder election procedures remain unchanged from [E-VPN].

### 7.3. Network Multi-homing

When an Ethernet network is multi-homed to a set of MES nodes running PBB-EVPN, an all-active redundancy model can be supported with per service instance (i.e. I-SID) load-balancing. In this model, DF election is performed to ensure that a single MES node in the redundancy group is responsible for forwarding traffic associated with a given I-SID. This guarantees that no forwarding loops are created. Filtering based on DF state applies to both unicast and multicast traffic, and in both access-to-core as well as core-to-access directions (unlike the multi-homed device scenario where DF filtering is limited to multi-destination frames in the core-to-access direction).

Similar to the multi-homed device scenario, a unique B-MAC address is used on the MES per multi-homed network (Segment). This helps eliminate the need for C-MAC address flushing in all but one failure scenario (more details on this in the Failure Handling section below). The B-MAC address may be auto-provisioned by snooping on the BPDUs of the multi-homed network: the B-MAC address is set to the root bridge ID of the CIST albeit with the ‘Locally Administered’ bit set.
7.3.1. B-MAC Address Advertisement

For every multi-homed network, the MES advertises two MAC Advertisement routes with different RDs and identical MAC addresses and ESIs. One of these routes will be tagged with a lower Local Pref attribute than the other. The route with the higher Local Pref will be tagged with the RTs corresponding to the I-SIDs for which the advertising MES is the DF. Whereas, the route with the lower Local Pref will be tagged with the RTs corresponding to the I-SIDs for which the advertising MES is the backup DF. Consider the example network of the figure below, where a multi-homed network (MHN1) is connected to two MES nodes (MES1 and MES2).

![Figure 3: Multi-homed Network](image)

Both MES nodes use the same B-MAC address (BM1) for the Ethernet Segment (ESI1) associated with MHN1. Assume, for instance, that MES1 is the DF for the even I-SIDs whereas MES2 is the DF for the odd I-SIDs. In this example, the routes advertised by MES1 and MES2 would be as follows:

**MES1:**
- Route 1: RD11, BM1, ESI1, Local Pref = 120, RT2, RT4, RT6...
- Route 2: RD12, BM1, ESI1, Local Pref = 80, RT1, RT3, RT5...

**MES2:**
- Route 1: RD21, BM1, ESI1, Local Pref = 120, RT1, RT3, RT5...
- Route 2: RD22, BM1, ESI1, Local Pref = 80, RT2, RT4, RT6

Upon receiving the above MAC Advertisement routes, the remote MES nodes (e.g. MESr) would install forwarding entries for BM1 towards MES1 for the even I-SIDs, and towards MES2 for the odd I-SIDs.

It is worth noting that the procedures of this section can also be used for a multi-homed device in order to support all-active redundancy with per I-SID load-balancing.

7.3.2. Failure Handling

In the case of an MES node failure, or when the MES is isolated from the multi-homed network due to a port or link failure, the affected
MES withdraws its MAC Advertisement routes for the associated B-MAC. This serves as a trigger for the remote MES nodes to adjust their forwarding entries to point to the backup DF. Because the same B-MAC address is used on both the DF and backup DF nodes, then there is no need to flush the C-MAC address table upon the occurrence of these failures.

In the case where the multi-homed network is partitioned, the MES nodes can detect this condition by snooping on the network’s BPDUs. When a MES detects that the root bridge ID has changed, it must change the value of the B-MAC address associated with the Ethernet Segment. This is done by the MES withdrawing the previous MAC Advertisement route, and advertising a new route for the updated B-MAC. The MES, which detects the failure, must inform the remote MES nodes to flush their C-MAC address tables for the affected I-SIDs. This is required because when the multi-homed network is partitioned, certain C-MAC addresses will move from being associated with the old B-MAC address to the new B-MAC addresses. Other C-MAC addresses will have their reachability remaining intact. Given that the MES node has no means of identifying which C-MACs have moved and which have not, the entire C-MAC forwarding table for the affected I-SIDs must be flushed. The affected MES signals the need for the C-MAC flushing by sending the MAC Mobility Extended Community in the MP_UNREACH_NLRI attribute containing the E-VPN NLRI for the withdrawn MAC Advertisement route.

7.4. Frame Forwarding

The frame forwarding functions are divided in between the Bridge Module, which hosts the [802.1ah] Backbone Edge Bridge (BEB) functionality, and the MPLS Forwarder which handles the MPLS imposition/disposition. The details of frame forwarding for unicast and multi-destination frames are discussed next.

7.4.1. Unicast

Known unicast traffic received from the AC will be PBB-encapsulated by the MES using the B-MAC source address corresponding to the originating site. The unicast B-MAC destination address is determined based on a lookup of the C-MAC destination address (the binding of the two is done via transparent learning of reverse traffic). The resulting frame is then encapsulated with an LSP tunnel label and the MPLS label which uniquely identifies the B-MAC destination address on the egress MES. If per flow load-balancing over ECMPs in the MPLS core is required, then a flow label is added as the end of stack label.

For unknown unicast traffic, the MES forwards these frames over MPLS core. When these frames are to be forwarded, then the same set of
options used for forwarding multicast/broadcast frames (as described in next section) are used.

7.4.2. Multicast/Broadcast

Multi-destination frames received from the AC will be PBB-encapsulated by the MES using the B-MAC source address corresponding to the originating site. The multicast B-MAC destination address is selected based on the value of the I-SID as defined in [802.1ah]. The resulting frame is then forwarded over the MPLS core using one out of the following two options:

Option 1: the MPLS Forwarder can perform ingress replication over a set of MP2P tunnel LSPs. The frame is encapsulated with a tunnel LSP label and the E-VPN ingress replication label advertised in the Inclusive Multicast Route.

Option 2: the MPLS Forwarder can use P2MP tunnel LSP per the procedures defined in [E-VPN]. This includes either the use of Inclusive or Aggregate Inclusive trees.

Note that the same procedures for advertising and handling the Inclusive Multicast Route defined in [E-VPN] apply here.

8. Minimizing ARP Broadcast

The MES nodes implement an ARP-proxy function in order to minimize the volume of ARP traffic that is broadcasted over the MPLS network. This is achieved by having each MES node snoop on ARP request and response messages received over the access interfaces or the MPLS core. The MES builds a cache of IP / MAC address bindings from these snooped messages. The MES then uses this cache to respond to ARP requests ingress on access ports and targeting hosts that are in remote sites. If the MES finds a match for the IP address in its ARP cache, it responds back to the requesting host and drops the request. Otherwise, if it does not find a match, then the request is flooded over the MPLS network using either ingress replication or LSM.

9. Seamless Interworking with TRILL and IEEE 802.1aq/802.1Qbp

PBB-EVPN enables seamless connectivity of TRILL or 802.1aq/802.1Qbp networks over an MPLS/IP core while maintaining control-plane separation among these networks. We will refer to one or any of TRILL, 802.1aq or 802.1Qbp networks collectively as ‘NG-Ethernet networks’ thereafter.

Every NG-Ethernet network that is connected to the MPLS core runs an independent instance of the corresponding IS-IS control-plane. Each MES participates in the NG-Ethernet network control plane of its local site. The MES peers, in IS-IS protocol, with the switches internal to the site, but does not terminate the TRILL / PBB data-
plane encapsulation. So, from a control-plane viewpoint, the MES appears as an edge switch; whereas, from a data-plane viewpoint, the MES appears as a core switch to the NG-Ethernet network. The MES nodes encapsulate TRILL / PBB frames with MPLS in the imposition path, and de-capitalize them in the disposition path.

9.1. TRILL Nickname Advertisement Route

A new BGP route is defined to support the interconnection of TRILL networks over PBB-EVPN: the TRILL Nickname Advertisement route, encoded as follows:

```
+---------------------------------------+
| RD (8 octets)                         |
+---------------------------------------+
| Ethernet Segment Identifier (10 octets)|
+---------------------------------------+
| Ethernet Tag ID (4 octets)            |
+---------------------------------------+
| Nickname Length (1 octet)             |
+---------------------------------------+
| RBridge Nickname (2 octets)           |
+---------------------------------------+
| MPLS Label (n * 3 octets)             |
+---------------------------------------+
```

Figure 4: TRILL Nickname Advertisement Route

The MES uses this route to advertise the reachability of TRILL RBridge nicknames to other MES nodes in the VPN instance. The MPLS label advertised in this route can be allocated on a per VPN basis and serves the purpose of identifying to the disposition MES that the MPLS-encapsulated packet holds an MPLS-encapsulated TRILL frame.

The encapsulation for the transport of TRILL frames over MPLS is encoded as shown in the figure below:

```
+------------------+
| IP/MPLS Header   |
+------------------+
| TRILL Header     |
+------------------+
| Ethernet Header  |
+------------------+
| Ethernet Payload |
+------------------+
| Ethernet FCS     |
+------------------+
```

Figure 5: TRILL over MPLS Encapsulation
It is worth noting here that while it is possible to transport Ethernet encapsulated TRILL frames over MPLS, that approach unnecessarily wastes 16 bytes per packet. That approach further requires either the use of well-known MAC addresses or having the MES nodes advertise in BGP their device MAC addresses, in order to resolve the TRILL next-hop L2 adjacency. To that end, it is simpler and more efficient to transport TRILL natively over MPLS and that is why we are defining the above BGP route for TRILL Nickname advertisement.

9.2. IEEE 802.1aq / 802.1Qbp B-MAC Advertisement Route

B-MAC addresses associated with 802.1aq / 802.1Qbp switches are advertised using the BGP MAC Advertisement route already defined in [E-VPN].

The encapsulation for the transport of PBB frames over MPLS is similar to that of classical Ethernet, albeit with the additional PBB header, as shown in the figure below:

```
+------------------+
| IP/MPLS Header   |
+------------------+
| PBB Header       |
+------------------+
| Ethernet Header  |
+------------------+
| Ethernet Payload |
+------------------+
| Ethernet FCS     |
+------------------+
```

Figure 6: PBB over MPLS Encapsulation

9.3. Operation

For correct connectivity, the TRILL Nicknames or 802.1aq/802.1Qbp B-MACs must be globally unique in the network. This can be achieved, for instance, by using a hierarchical Nickname (or B-MAC) assignment paradigm, and encoding a Site ID in the high-order bits of the Nickname (or B-MAC):

Nickname (or B-MAC) = [Site ID : Rbridge ID (or MAC)]

The only practical difference between TRILL Nicknames and B-MACs, in this regards, is with respect to the size of the address space: Nicknames are 16-bits wide whereas B-MACs are 48-bits wide.
Every MES then advertises (in BGP) the Nicknames (or B-MACs) of all switches local to its site in the TRILL Nickname Advertisement routes (or MAC Advertisement routes).
Furthermore, the MES advertises in IS-IS (to the local island) the Rbridge nicknames (or B-MACs) of all remote switches in all the other TRILL (or IEEE 802.1aq/802.1Qbp) islands that the MES has learned via BGP.

Note that by having multiple MES nodes (connected to the same TRILL or 802.1aq /802.1Qbp island) advertise routes to the same RBridge nickname (or B-MAC), with equal BGP Local_Pref attribute, it is possible to perform active/active load-balancing to/from the MPLS core.

When a MES receives an Ethernet-encapsulated TRILL frame from the access side, it removes the Ethernet encapsulation (i.e. outer MAC header), and performs a lookup on the egress RBridge nickname in the TRILL header to identify the next-hop. If the lookup yields that the next hop is a remote MES, the local MES would then encapsulate the TRILL frame in MPLS. The label stack comprises of the VPN label (advertised by the remote MES), followed by an LSP/IGP label. From that point onwards, regular MPLS forwarding is applied.

On the disposition MES, assuming penultimate-hop-popping is employed, the MES receives the MPLS-encapsulated TRILL frame with a single label: the VPN label. The value of the label indicates to the disposition MES that this is a TRILL packet, so the label is popped, the TTL field (in the TRILL header) is reinitialized and normal TRILL processing is employed from this point onwards.

By the same token, when a MES receives a PBB-encapsulated Ethernet frame from the access side, it performs a lookup on the B-MAC destination address to identify the next hop. If the lookup yields that the next hop is a remote MES, the local MES would then encapsulate the PBB frame in MPLS. The label stack comprises of the VPN label (advertised by the remote PE), followed by an LSP/IGP label. From that point onwards, regular MPLS forwarding is applied.

On the disposition MES, assuming penultimate-hop-popping is employed, the MES receives the MPLS-encapsulated PBB frame with a single label: the VPN label. The value of the label indicates to the disposition MES that this is a PBB frame, so the label is popped, the TTL field (in the 802.1Qbp F-Tag) is reinitialized and normal PBB processing is employed from this point onwards.

10. Solution Advantages

In this section, we discuss the advantages of the PBB-EVPN solution in the context of the requirements set forth in section 3 above.
10.1. MAC Advertisement Route Scalability

In PBB-EVPN the number of MAC Advertisement Routes is a function of the number of segments (sites), rather than the number of hosts/servers. This is because the B-MAC addresses of the MESes, rather than C-MAC addresses (of hosts/servers) are being advertised in BGP. And, as discussed above, there’s a one-to-one mapping between multi-homed segments and B-MAC addresses, whereas there’s a one-to-one or many-to-one mapping between single-homed segments and B-MAC addresses for a given MES. As a result, the volume of MAC Advertisement Routes in PBB-EVPN is multiple orders of magnitude less than E-VPN.

10.2. C-MAC Mobility with MAC Sub-netting

In PBB-EVPN, if a MES allocates its B-MAC addresses from a contiguous range, then it can advertise a MAC prefix rather than individual 48-bit addresses. It should be noted that B-MAC addresses can easily be assigned from a contiguous range because MES nodes are within the provider administrative domain; however, CE devices and hosts are typically not within the provider administrative domain. The advantage of such MAC address sub-netting can be maintained even as C-MAC addresses move from one Ethernet segment to another. This is because the C-MAC address to B-MAC address association is learnt in the data-plane and C-MAC addresses are not advertised in BGP. To illustrate how this compares to E-VPN, consider the following example:

If a MES running E-VPN advertises reachability for a MAC subnet that spans N addresses via a particular segment, and then 50% of the MAC addresses in that subnet move to other segments (e.g. due to virtual machine mobility), then in the worst case, N/2 additional MAC Advertisement routes need to be sent for the MAC addresses that have moved. This defeats the purpose of the sub-netting. With PBB-EVPN, on the other hand, the sub-netting applies to the B-MAC addresses which are statically associated with MES nodes and are not subject to mobility. As C-MAC addresses move from one segment to another, the binding of C-MAC to B-MAC addresses is updated via data-plane learning.

10.3. C-MAC Address Learning and Confinement

In PBB-EVPN, C-MAC address reachability information is built via data-plane learning. As such, MES nodes not participating in active conversations involving a particular C-MAC address will purge that address from their forwarding tables. Furthermore, since C-MAC addresses are not distributed in BGP, MES nodes will not maintain any record of them in control-plane routing table.

10.4. Seamless Interworking with TRILL and 802.1aq Access Networks
Consider the scenario where two access networks, one running MPLS and the other running 802.1aq, are interconnected via an MPLS backbone network. The figure below shows such an example network.

![Diagram](image)

Figure 7: Interoperability with 802.1aq

If the MPLS backbone network employs E-VPN, then the 802.1aq data-plane encapsulation must be terminated on MES1 or the edge device connecting to MES1. Either way, all the MES nodes that are part of the associated service instances will be exposed to all the C-MAC addresses of all hosts/servers connected to the access networks. However, if the MPLS backbone network employs PBB-EVPN, then the 802.1aq encapsulation can be extended over the MPLS backbone, thereby maintaining C-MAC address transparency on MES1. If PBB-EVPN is also extended over the MPLS access network on the right, then C-MAC addresses would be transparent to MES2 as well.

Interoperability with TRILL access network will be described in future revision of this draft.

10.5. Per Site Policy Support

In PBB-EVPN, a unique B-MAC address can be associated with every site (single-homed or multi-homed). Given that the B-MAC addresses are sent in BGP MAC Advertisement routes, it is possible to define per site (i.e. B-MAC) forwarding policies including policies for E-TREE service.

10.6. Avoiding C-MAC Address Flushing

With PBB-EVPN, it is possible to avoid C-MAC address flushing upon topology change affecting a multi-homed device. To illustrate this, consider the example network of Figure 1. Both MES1 and MES2 advertise the same B-MAC address (BM1) to MES2. MES2 then learns the C-MAC addresses of the servers/hosts behind CE1 via data-plane learning. If AC1 fails, then MES3 does not need to flush any of the C-MAC addresses learnt and associated with BM1. This is because MES1 will withdraw the MAC Advertisement routes associated with BM1,
by leading MES3 to have a single adjacency (to MES2) for this B-MAC address. Therefore, the topology change is communicated to MES3 and no C-MAC address flushing is required.

11. Acknowledgements
    TBD.

12. Security Considerations
    There are no additional security aspects beyond those of VPLS/H-VPLS that need to be discussed here.

13. IANA Considerations
    This document requires IANA to assign a new SAFI value for L2VPN_MAC SAFI.

    This document is being submitted for use in IETF standards discussions.

15. Normative References

16. Informative References

17. Authors’ Addresses
    Ali Sajassi
    Cisco
    170 West Tasman Drive
    San Jose, CA  95134, US
    Email: sajassi@cisco.com
    Samer Salam

    Sajassi, et al.
Coordinated Multicast Trees (CMT) for TRILL
draft-tissa-trill-cmt-00.txt

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on July 5, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
Internet-Draft  Coordinated Multicast Trees for TRILL  January 2012

carefully, as they describe your rights and restrictions with respect to this document.

Abstract

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

Table of Contents

1. Introduction...................................................3
   1.1. Contributors..............................................4
2. Conventions used in this document..............................5
3. IS-IS extension................................................5
   3.1. AFFINITY TLV..............................................5
4. Multicast Tree Construction and use of Affinity Sub-TLV....7
   4.1. Update to RFC 6325........................................8
   4.2. Announcing virtual RBridge nickname.......................9
   4.3. Affinity Sub-TLV capability...............................9
5. Theory of operation............................................9
   5.1. Distribution Tree provisioning............................9
   5.2. Affinity Sub-TLV advertisement...........................9
   5.3. Affinity sub-TLV conflict resolution.....................10
   5.4. Ingress Multi-Destination Forwarding.....................10
   5.4.1. Forwarding when n < k...............................10
   5.5. Egress Multi-Destination Forwarding......................11
   5.5.1. Traffic arriving on an assigned Tree to RBk-RBv.....11
   5.5.2. Traffic arriving on other Trees......................11
   5.6. Failure scenarios........................................11
   5.6.1. Edge RBridge RBk failure.............................11
   5.7. Backward compatibility..................................12
6. Security Considerations.......................................12
7. IANA Considerations...........................................13
8. References....................................................13
   8.1. Normative References....................................13

Senevirathne             Expires July 5, 2012                  [Page 2]
1. Introduction

TRILL presented in [RFC6325] and other related documents, provide methods of utilizing all available paths for active forwarding, with minimum configuration. TRILL utilizes IS-IS as control plane and encapsulates native frames with a TRILL header.

Legacy networks utilize IEEE 802.1D Spanning Tree Protocol as the control protocol and utilizes at any given time, a single path among all available paths for active forwarding. Legacy networks forward frames in "native" format.

[RFC6325],[RFC6327] and [RFC6439] provide methods for interoperability between TRILL and Legacy networks. [RFC6439], provide active-standby solution, where only one of the R Bridges is in active forwarding state for any given VLAN. The R Bridge in active forwarding state for any given VLAN is referred to as the Appointed Forwarder (AF). All frames ingressing into a TRILL network via the Appointed Forwarder are encapsulated with the TRILL header with a nickname held by the ingress AF R Bridge. Due to failures, reconfigurations and other network dynamics, Appointed Forwarder for any set of VLANs may change. R Bridges maintain forwarding table that contain destination MAC address to egress R Bridge binding. In the event of AF change, forwarding tables of remote R Bridges may continue to forward traffic to the previous AF and may get discarded at the egress, causing traffic disruption.

Mission critical applications such as High Performance Data Centers require resiliency during failover. The active-active forwarding model minimizes impact during failures and maximizes the available network bandwidth. A typical deployment scenario, depicted in Figure 2, which may have either End Stations and/or Legacy bridges attached to the R Bridges. These Legacy devices typically are multi-homed to several R Bridges and treats all of the uplinks as a single Link Aggregation (LAG) bundle. The Appointed Forwarder designation presented in [RFC6439] requires each of the edge R Bridges to exchange TRILL hello packets. By design, a LAG does not forward packets received on one of the member ports of the LAG to other member ports of the same LAG. As a result AF designation methods presented in [RFC6439] cannot be applied to deployment scenario depicted in Figure 2.
An active-active load sharing model can be implemented by representing the edge of the network connected to a specific group of RBridges by a single virtual RBridge. In addition to an active-active forwarding model, there may be other applications that may require similar representations.

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

In this document we present a method that allows RBridges to specify the path of association to distribution trees. Remote RBridges calculate the SPF and derive the RPF for distribution trees based on the distribution tree association advertisements. In the absence of distribution tree association advertisements, remote RBridges derive the SPF based on the algorithm specified in section 4.5.1 of [RFC6325].

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

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

1.1. Contributors

The work in this document is a result of much passionate discussions and contributions from following individuals. Their names are listed in alphabetical order and do not in any means intended to be ranked in any other way.
2. Conventions used in this document

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

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

3. IS-IS extension

3.1. AFFINITY TLV

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

AFFINITY TLV is a sub-TLV under Router capability TLV (242) [RFC 4971]. [RFC6326bis] formally defines the Affinity sub-TLV. The details of the Affinity sub-TLV are presented for clarity and readers are encouraged to refer to [RFC6326bis] for latest updates.
Figure 1 Affinity TLV structure

- Type: AFINITYTLV, (TBD).
- Length: 4 + 2*n, where n is the number of nicknames of affinity trees listed tree numbers.
- Nickname: TRILL 16bit nickname of the RBridge whose association to the listed multicast trees are through the announcing RBRidge.
- Number of trees: number of trees for which association (affinity) being announced.
- Tree-num of preferred root: The tree Number of the multicast tree.
4. Multicast Tree Construction and use of Affinity Sub-TLV

![Reference Topology Diagram](image)

Figure 2 Reference Topology
4.1. Update to RFC 6325

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

Each RBridge that desires to be a parent RBridge for a specific multi-destination distribution tree x for child RBridge RBy or nickname Ny (in the event of announcing multiple nicknames), announces the desired association through Affinity sub-TLV.

When such an Affinity sub-TLV is present, the association specified by the affinity sub-TLV MUST be used when constructing the SPF tree. In the absence of such Affinity sub-TLV, or if there are RBRidges in the network that are do not support Affinity sub-TLV, SPF tree is calculated as specified in the section 4.5.1 of [RFC6325]. Section 4.3. below explains methods of identifying RBRidges that support Affinity sub-TLV capability.
4.2. Announcing virtual RBridge nickname

Each edge RBridge RB1 to RBk advertise virtual RBridge nickname RBv using nickname sub-TLV (6), [RFC6326bis], along with their original nicknames RBi, RBj.

4.3. Affinity Sub-TLV capability.

RBridges that announce the TRILL version sub-TLV [RFC6326bis] and set the Affinity capability bit (section 7.) support the Affinity sub-TLV and calculation of multi-destination distribution trees as specified herein.

5. Theory of operation

5.1. Distribution Tree provisioning

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

If n >= k

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

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

If n < k

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

5.2. Affinity Sub-TLV advertisement

Each RBridge in the RB1..RBk domain advertises Affinity TLV on behalf of RBv.

As an example, let’s assume that RB1 has chosen Trees t1 and tk+1 on behalf of RBv.
RB1 advertises affinity TLV; (RBv, Num of Trees=2, t1, tk+1.  

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

5.3. Affinity sub-TLV conflict resolution

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

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

Similarly, remote RBridges MUST honor the Affinity sub-TLV from the RBridge with the highest priority to be a tree root and ignore the conflicting Affinity sub-TLV entries advertised by the RBridges with lower priorities to be tree roots.

5.4. Ingress Multi-Destination Forwarding

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

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

5.4.1. Forwarding when n < k

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

1. This RBridge should stop forwarding frames from the CE nodes, and should mark its link as passive. This will prevent CE nodes
from forwarding data on to this RBridge, and only use those
RBridges which have been assigned a tree -OR-
2. This RBridge tunnels multi-destination frames received from
attached native devices to an RBridge RBv that has an assigned
tree. The tunnel destination should forward it to the TRILL
network, and also to its local access links. (The mechanism
of tunneling and handshake between the tunnel source and
destination are out of scope of this specification and may be
addressed in future documents).

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

5.5. Egress Multi-Destination Forwarding

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

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

5.5.2. Traffic Arriving on other Trees

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

5.6. Failure scenarios

5.6.1. Edge RBridge RBk failure

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

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

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

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

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

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

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

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

5.7. Backward compatibility

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

Example:

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

6. Security Considerations

Security considerations are similar to RFC 6325, RFC 6326 and RFC 6327. Additional security considerations are being discussed.
7. IANA Considerations

IANA is requested to allocate a capability bit for "Affinity Supported" in the TRILL-VER sub-TLV. "Affinity Supported" capability bit and Affinity sub-TLV are specified and allocated in [RFC6326bis].

8. References

8.1. Normative References


8.2. Informative References


9. Acknowledgments

Authors wish to extend their appreciations towards individuals who volunteered to review and comment on the work presented in this document and provided constructive and critical feedback. Specific acknowledgements are due for Ronak Desai, Gayatri Ramachandran, Varun Shah and Anoop Ghanwani.

This document was prepared using 2-Word-v2.0.template.dot.
10. Authors’ Addresses

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

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

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

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

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

Email: jon.hudson@gmail.com
Default Nickname Based Approach for Multilevel TRILL
draft-tissa-trill-multilevel-00.txt

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on August 21, 2012.
Abstract

Multilevel TRILL allows the interconnection of multiple TRILL networks to form a larger TRILL network without proportionally increasing the size of the IS-IS LSP DB. In this document, an approach based on default route concept is presented. Also, presented in the document is a novel method of constructing multi-destination trees using partial nickname space. Methods presented in this document are compatible with the RFC6325 specified data plane operations.

Table of Contents

1. Introduction ................................................... 3
2. Conventions used in this document .............................. 4
3. Solution Overview .............................................. 4
4. Operational Overview ........................................... 5
  4.1. Unicast Forwarding ........................................ 5
  4.2. IS-IS Protocol changes for unicast forwarding ............. 5
  4.3. MAC Address Learning ...................................... 6
  4.4. Multicast ................................................. 6
  4.5. Life of Multicast frame ................................... 9
5. Area Affinity sub-TLV ......................................... 10
6. Nickname acquisition and conflict resolution ................. 11
  6.1. Nickname Management sub-TLV .............................. 13
7. Further optimizations .......................................... 14
  7.1. Leaking of TRILL IS-IS sub-TLV within areas .............. 14
  7.2. Identification of Global Trees ................................ 15
     7.2.1. Global Tree capability sub-TLV ........................... 17
     7.2.2. Global Tree proposal sub-TLV ........................... 17
     7.2.3. Global Tree Identifier sub-TLV ........................... 18
The TRILL Base Protocol Specification [RFC6325] provides a method for forwarding Layer 2 data frames over multiple active links, thereby optimizing network bandwidth and resiliency. TRILL requires native Layer 2 frames to be encapsulated with the TRILL header. TRILL devices (RBridges) are identified with a 16 bit identifier called a nickname. The TRILL header contains egress and ingress RBridge nicknames. Intermediate RBridges perform forwarding based on the egress nickname. TRILL utilize the IS-IS protocol to distribute RBridge nicknames.

TRILL Base Protocol Specification [RFC6325] specifies a tree based paradigm to forward broadcast and multicast traffic as well as unknown unicast traffic.

Traditional Layer 2 devices perform forwarding based on MAC addresses. As a result, in theory, all of the devices in the network are required to learn all of the MAC addresses in the Layer 2 domain. This leads to very large forwarding table sizes in the devices which limits the size of the layer 2 domain. Forwarding within the TRILL network occurs not based on MAC addresses but based on the RBridge nicknames. Hence, TRILL based networks have significant potential to be the core forwarding plane of very large datacenters.

Large datacenters are often multisite in nature and contain a large number of RBridges. TRILL is designed to be a single IS-IS area. As the size of the TRILL network grows, the size of the IS-IS LSP database grows, leading to network convergence delays and increased volatility during transient conditions such as link flaps.
As mentioned above TRILL utilizes a tree based forwarding paradigm for multi-destination traffic. In large TRILL networks this may lead to sub-optimal forwarding. Additionally, entire network wide multicasting may lead to network bandwidth inefficiencies and have a negative impact on performance.

In order to support scaling and performance of large TRILL networks, it is important to have methods to:

1. Limit the size of the IS-IS LSP database
2. Optimize multicast forwarding
3. Limit the scope of flooding and broadcast traffic

In this document we propose methods that allow implementing multi-level TRILL without any data plane changes as well as meet the above design goals.

Also presented in this document is a novel method of constructing multi-destination trees using partial nickname space.

2. Conventions used in this document

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

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

3. Solution Overview

Herein we present a high level view of the proposed solution; differing the details of the solution to subsequent sections.

The TRILL campus is divided into multiple IS-IS L1 Areas interconnected by L2 backbone area. The backbone area may be interconnected by an IS-IS K2 area or by some other means. For example, one does not preclude a configuration based approach for the L2 backbone.

Area border RBridges indicate their ability to reach other areas by setting the Attach bit in IS-IS Link State PDU.

R Bridges forward frames destined to RBridges in the area using the exact match of nicknames. Frames destined to RBridges outside of the
L1 area are forwarded using the default route advertised by the area border RBridges.

Campus wide Multi-destination trees are partitioned into two parts. A backbone tree rooted on the L2 backbone and local trees rooted within each L1 area. For campus wide trees the local trees and the backbone tree have the same nickname. This avoids the need for egress RBridge nickname translation at the border RBridges).

One of the border RBridges performs connecting its local tree to the corresponding backbone tree. The Affinity sub-TLV and Area Affinity sub-TLV information facilitate constructing proper RPF states.

4. Operational Overview

4.1. Unicast Forwarding

The TRILL campus is divided into multiple IS-IS L1 Areas with a Layer 2 backbone. (Figure 1).

The "Attached" bit in IS-IS PDU is used to indicate the advertising IS connected to other areas. In this document we propose to leverage the "Attached" bit to identify the border RBridges and forward traffic for all un-resolved nicknames to the border RBridges.

Border RBridges possess the complete nickname space of the TRILL campus. Utilizing this information, ingress area border RBridge forward TRILL frames to the egress area border RBridge via the L2 backbone.

Egress area border RBridges, forward the frame normally to the intended destination in the given L1 Area.

4.2. IS-IS Protocol changes for unicast forwarding

Support for non zero IS-IS areas for TRILL.

No new TLV or sub-TLV required.

Border RBridges advertise LSP with the "Attached" bit set

L1 Area RBridges are required to advertise TRILL related sub-TLV defined in [RFC 6326] with the router capability bit "S" set. The capability bit "D" MUST be set to zero (0). This allows leaking L1 PDU to the L2 backbone area but not to other L1 areas [RFC4971].
4.3. MAC Address Learning

Egress RBridge learn remote MAC addresses against the actual nickname of the ingress RBRidge.

As an example: Let’s Assume MAC address A is attached to RB1 and MAC address B is attached to RB7.

RB1 receives a frame destined to MAC B. RB1 does not know the location of MAC B. However, local policy such as, port or VLAN indicates that the frame is of global scope. RB1 transmits the frame on global tree "t" as an unknown unicast frame.

RB7 receives the frame and learn MAC A is associated with RB1. RB7 programs its forwarding tables such that MAC-A is associated with RB1. It also forwards the frame to MAC-B.

MAC-B responds. RB7 has the association of MAC-A to RB1. Hence, RB7 forwards the frame to RB1 as a unicast frame, with egress RBridge nickname as RB1 and ingress RBridge nickname as RB7.

The frame follows the normal unicast forwarding process explained above and arrives at RB1. RB1, learns the MAC-B association to RB7 and updates its forwarding tables accordingly. Also, RB1 forwards the frame to MAC-A.

4.4. Multicast

Multicast forwarding has two parts: 1. Construction of the multicast trees 2. RPF (Reverse Path Forwarding) check.

In most of the real world deployments, not all of the traffic is required or desired to span across the entire TRILL campus. The majority of the traffic tends to have a local scope and some subset of traffic to have a global scope.

The scope of global traffic may be identified either through VLAN or via finegrain label that spans across the entire TRILL campus.

In this document we propose to classify TRILL multi-destination trees into two types:

1. Local trees (trees that have a scope within the local area)
2. Global trees (trees that have a scope throughout the TRILL campus)
Multi-destination traffic of local scope is forwarded using Local trees. Multi-destination traffic of global scope is forwarded using global trees.

Construction of global multi-destination trees and performing RPF check for such trees requires knowledge of all of the RBRidges in the entire TRILL campus. In a large TRILL campus, construction of such global trees that need information of all RBRidges may not only lack scalability but also may run into instabilities during network changes. Additionally, when the TRILL campus is divided into multiple IS-IS L1 areas, RBRidges within an L1 areas do not possess reachability information for other areas. Thus, constructing such global trees may not be possible.

In this document we propose an [RFC6325] compatible approach of building multicast trees to address the issues mentioned above.

In the proposed method, global trees (campus wide trees) are partitioned into two instances; a backbone tree instance and set of local tree instance per each area.

Backbone tree – An instance of the tree rooted in the L2 backbone. The Backbone tree is represented by the same nickname as the global tree.

Local tree – An instance of a tree per area, per campus wide tree, rooted within each area. Each instance of the tree in each area is represented by the same nickname as the global tree. (This is important to avoid the need for tree translation at the border RBRidges)

L2 LSPs advertise the backbone tree.

L1 LSPs advertise the corresponding local-tree within each area.

One of the L1-L2 area border RBRidges in a given area is assigned the role of Rendezvous Point (RP) for the specific local tree (more details are presented in section 8.).

Each RP functions as the plumb between the global tree and local tree. Both the trees have exactly the same nickname.

RP An RP advertises affinity to the rest of the campus for the specified tree by using the Affinity sub-TLV [trillcmt]. In the Affinity TLV associated nickname MUST be specified as zero to indicate the Affinity is related to default route. We refer to this
usage of Affinity TLV as the default Affinity sub-TLV in the rest of the document.

Each RBridge in the local L1 area builds its multi-destination SPF tree as specified in [RFC6325] and [trillcmt]. RPF checks are performed as specified in [RFC6325] and [trillcmt]. Additionally, RPF checks for default nicknames (i.e. all unknown nicknames to the local L1 area) are performed per the association specified by the default Affinity sub-TLV.

Additionally, each RP on behalf of the local Area it is representing for multi-cast tree Tx, advertises Area Affinity-TLV towards the L2 backbone area. The Area Affinity TLV, include the L1 Area ID of the associated area. The Area Affinity TLV, notifies RBridges in the L2 area to enable the RPF check to accept nicknames in the associated L1 area from the announcing RP. The Area Affinity TLV allows greater scaling of the IS-IS LSP DB. If Affinity TLV contains all of the nicknames the IS-IS PDU size increases. Use of the Area Affinity sub-TLV to summarize the entire area in a single sub-TLV, limits the size of the LSP DB as well as PDU size. (Please see below for the structure of the Area Affinity sub-TLV).

![Figure 1 Sample Topology](image-url)
Above Figure 1 depicts a sample Topology, Figure 2 depicts the corresponding logical topology. Local multicast trees and the global multicast trees have the same nickname "t". "X" denote the branches of the multicast tree that are not used for multicast traffic reception or transmission. L1 RBridges, RB1, RB7 install RPF check based on the default Affinity TLV. L1 area border RBridges MUST not install the default RPF check on L2 interfaces, instead they MUST honor the Area Affinity TLV and install explicit RPF checks. Strict RPF check on area border RBridges are mandatory to prevent transient loops during topology changes.

4.5. Life of Multicast frame.

RB1 ingresses multicast frames of global scope to global tree "t". The multicast frame traverses the tree "t" and arrives at RBridges RB2 and RB3. RB2 is the Rendezvous Point (RP) for tree "t" for Area A1. RB2 forward the frame to backbone multicast tree "t" as well as to the other local ports. RB3 is not the RP for tree "t" for area A1. Hence, it forwards the frame to local ports but does not forward along the backbone multicast tree "t".

The multicast frame traverses along the backbone tree "t" and arrives at Area A2 border RBridges RB5 and RB6. The RB6, which is not the Rendezvous Point (RP) for backbone tree "t" for Area A2, accepts multi-destination frames arriving on L2 interface and only
forwards to other applicable L2 interfaces. Non RP RBridges do not forward multi-destination frames arriving on global tree on L1 area interface. RB5, the RP for the Area A2 for the multicast tree "t", contains the RPF check to accept frames from others areas on tree "t" on its L2 area interface. Hence, RB5, accept the multicast frame from tree "t" and forward along the local tree "t" and its local ports.

The multicast packet traverses along the local tree "t" in Area A2 and arrive at RB7. RB7 contains RPF check installed based on the Default Affinity TLV for tree "t", to receive multicast traffic for tree "t" that arrives from the RP facing interface. RB7 accepts the multicast frame arriving on local tree "t" with ingress RBridge nickname RB1 and performs applicable forwarding as specified in [RFC6325].

RB6 is an Area border RBridge for Area A2, but not RP for the area. RB6 receives the multicast frame through the local tree "t". Non RP area border RBridges for RPF and multi-destination forwarding purposes function like a L1 area RBridge. RB6, honors the default Affinity TLV received from RB5 for local multicast tree "t". Hence, it installs RPF check to accept all nicknames (default nickname) for tree "t" from the L1 Area interface pointing towards RB5.

RB6 accepts the incoming multicast frame along tree "t" with the ingress RBridge nickname RB1 and performs applicable forwarding to locally attached ports. RB6, which is a non RP for tree "t" for area A2, MUST not forward the multicast frame to the global tree "t".

5. Area Affinity sub-TLV

Area Affinity sub-TLV is a sub-TLV under IS-IS Router capability TLV and contains the following structure.
Figure 3 Area Affinity TLV structure

- Type: AREA-AFF, (TBD).
- Length: variable. Length is 1 + length of Area ID + 2*n, where n is the number listed tree numbers.
- Area-ID: (variable length) is the IS-IS Area ID. Length can be 1-13 bytes long.
- Number of trees: number of trees for which association (affinity) being announced.
- Tree-num of preferred root: The tree Number of the multicast tree.

Area-affinity conflicts MUST be resolved using methods specified in [trillcmt].

6. Nickname acquisition and conflict resolution

In the proposed method, nicknames of RBridges in remote L1 areas are not advertised into the local L1 area by area border RBridges. However, L2 backbone RBridges and L1-L2 border RBridges contain the nicknames used by all the RBridges in the campus. We propose to
introduce a new IS-IS sub-TLV under Router capability TLV to
distribute available nickname space. New IS-IS sub TLV is Nickname
Management sub-TLV.

Nickname Management sub-TLV is announced by the area border RBridges
in to the local L1 area with Router capability bits D set to one and
S set to one. Nickname Management sub-TLV is announced in to L2 area
by the area border RBridges with S and D bit clear. These settings
ensure Nickname Management TLV is confined to the local L1 area L2
area and does not leak to other L1 areas.

Nickname Management sub-TLV instance announced in to the local area,
contains two sets of ranges. Local nickname ranges and dynamic
nickname ranges. Local nickname ranges are one or more sets of
ranges that network administrator has configured on the border
RBridges. Multiple local nickname ranges allow network administrator
to configure multiple sets of non contiguous nickname ranges.

L1 area RBridges SHOULD, first, select a nickname or nicknames from
the local ranges.

If entire local nickname space has exhausted (i.e. taken up by other
RBridges), then L1 area RBridges SHOULD select a nickname or
nicknames from the dynamic ranges.

It is recommended that RBridges use different nickname priorities to
differentiate nickname acquired by different methods. Nickname
priorities are assigned based on the acquisition method such that
configured nicknames have highest priority, followed by nicknames
derived from the local range, followed by nicknames derived from the
dynamic range.

Dynamic ranges are derived by the area border RBridges based on the
local nickname ranges of its and other areas. As an example let’s
assume area A1 has local nickname range of 100-200, A2 has a local
nickname range of 201-300. Then the dynamic range is from 1-99 and
301 to 65471 (0xFFBF). Nickname values 0 and 0xFFC0 to 0xFFFF are
reserved and MUST not be included in the dynamic nickname ranges.

Nickname Management sub-TLV instance announced in to the L2 backbone
area, contains only the local nickname ranges. Local nickname ranges
of each area allow other areas to derive applicable dynamic ranges
to announce in to the corresponding L1 area.
6.1. Nickname Management sub-TLV

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Type=NICK-MGMT | (1 byte) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Length | (1 byte) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Area-ID-Length | (1 byte) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Area ID | (variable 1..13) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| NL | (1 byte) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| starting nickname for l-range 1 | (2 bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| end nickname for l-range 1 | (2 bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| starting nickname for l-range n | (2 bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| end nickname for l-range n | (2 bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| ND | (1 byte) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| starting nickname for d-range 1 | (2 bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| end nickname for d-range 1 | (2 bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| starting nickname for d-range n | (2 bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| end nickname for d-range n | (2 bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 4 Nickname Management sub-TLV structure

- Type: NICK-MGMT, (TBD).
o Length: variable. Length is 1 + length of Area ID + 1+2*2*NL + 1 + 2*2*ND, where NL is the number local ranges and ND is number of dynamic ranges.

o Area-ID : (variable length) Is the IS-IS Area ID. Length can be 1-13 bytes long.

o NL : number of local nickname ranges.

o ND : number of dynamic ranges.

o Starting nickname for l-range n: starting nickname of the local range n.

o End nickname for l-range n: End nickname of the local range n.

o Starting nickname for d-range n: starting nickname of the dynamic range n.

o End nickname for d-range n: End nickname of the dynamic range n.

NOTE: Multiple instances of the nickname management sub-TLV MAY be included. Nickname management sub-TLV has usage in non multi-level deployments as well. Nickname management sub-TLV allows administrator to control nickname acquisition by RBridges.

7. Further optimizations

RFC6325 allows multiple instance of nickname sub-TLV. If more specific forwarding is required, for some critical RBridges, such nicknames MAY be advertised in a separate Router capability TLV, with S bit set. So it leaks to all L1 areas.

7.1. Leaking of TRILL IS-IS sub-TLV within areas

At the boundary nodes, the following information needs to be leaked from the Level-1 database to the Level-2 database. The nickname TLVs that are learned from all nodes within the same site needs to be redistributed to the Level-2 database. All such redistributed nickname TLVs will have the root priority set to 0. Note, that all border area nodes will announce this TLV into the Level-2 database. As a result, all Level-2 nodes will be able to see the reachability of all other nodes. This enables unicast traffic flow. With respect to multicast, redistributed nicknames are not to be used in the root election for global trees. The roots of the global trees will be from the set of nicknames that are in the Level-2 database. Once the
roots have been identified, these nicknames need to be leaked back to the Level-1 areas. Calculation of multi-destination trees are presented in section 7.2.

7.2. Identification of Global Trees

It is expected only a sub-set of traffic requires a global reach (inter Area). Majority of the traffic will be of local scope (intra area). Scope of the traffic can be identified either based on VLAN or fine-grain label. Traffic of global scope is forwarded using global trees. The trees of global scope may be indentified:

1. By means of configuration
2. By means of multi-destination tree announcements sub-TLV.

Point number 2 above requires further clarifications. We propose to introduce 3 new sub-TLV under IS-IS router capability TLV to advertise global multi-destination trees. These new sub-TLV are listed below. Format of the new sub-TLVs are presented in section 7.2.1. to 7.2.3.

- The Global Tree capability sub-TLV
- The Global Tree proposal sub-TLV
- The Global Tree identifier sub-TLV

Each RBridge announces two sets of capabilities; global tree capabilities and local tree capabilities. It announces, maximum number of global distributions trees it can compute. RBridges utilize Global Tree sub-TLV for the purpose of announcing its global tree capability. RBridges announce local tree capabilities using the Tree sub-TLV [RFC6326]. Global tree space is disjoint from the local tree space and MUST not have any effect on each other. Please refer to [RFC6325] for the process of how local trees are derived. In this section we present how the total number of global trees needed is calculated and identification of nickname of each tree root.

Each area border RBridge, using the global tree sub-TLV received from RBridges in its local area, derives the number of trees the area can support. The number of global trees "s", a local area can support is the fewest number of global trees that an RBridge in local area can support. Area border RBridges advertise in to the backbone, using the global Tree capability sub-TLV, the number of trees "s" that the given area can calculate. Global Tree capability sub-TLV announced in to the L2 backbone are advertised with "S" and "D" flags of Router Capability TLV set to 0. This prevent them leaking to L1 areas.
Rbridges in the L2 backbone calculate the number of global trees the campus can calculate. This number "i" is the fewest number of global trees among all areas.

Each RBridge in the L2 backbone using Global Tree proposal sub-TLV advertise the number of trees it want other RBriges in the L2 backbone to calculate. RBriges in the L2 backbone identifies number of global trees they need to calculate based on the number of trees "k" advertised by the highest priority RBridge in the L2 backbone.

The L2 area RBridge with the highest priority advertises set of nicknames for the global tree roots. These tree roots are selected based on tree root priority announced by L2 backbone RBriges. Global Tree Identifier sub-TLV is used for the purpose.

BSR RBRidge of each L1 area advertises nicknames of global tree roots using Global Tree Identifier sub-TLV in to the corresponding local area. BSR also advertises the number of Global trees k the local area needs to calculate using Global tree proposal sub-TLV.

RBriges in the local area contain only nicknames of the local area. Global Tree Identifier sub-TLV announced by the BSR contains nicknames that are unknown to the local L1 area RBriges. How do local RBriges calculate it SPF?

Global Tree Identifier sub-TLV contain nickname of the global trees ordered in ascending order. Default Affinity TLV announced by RP RBridge contains the tree-id(s) that it is an RP. Tree-id k in the Default affinity TLV corresponds to nickname k in the Global Tree Identifier. RBriges in the local L1 area calculate its SPF tree assuming the tree-k is rooted at the RP RBridge announcing the Default affinity sub-TLV for that tree.

Global Tree proposal sub-TLV and Global Tree Identifier sub-TLV MUST only be advertised by BSR. Sub-TLV from highest priority RBRidge is chosen, in the event of multiple RBriges advertising conflicting sub-TLVs.
7.2.1. Global Tree capability sub-TLV

```
+---------------------+
| Type = GL-TREES     | (1 byte)
+---------------------+
| Length              | (1 byte)
+---------------------+
| Maximum trees able to compute | (2 byte)
```

Figure 5 Global Tree Capability sub-TLV

Type: Router Capability sub-TLV type, TBD

Length: 2.

Maximum trees able to compute: An unsigned 16-bit integer indicates maximum number of global trees the announcing RBridge able to compute.

7.2.2. Global Tree proposal sub-TLV

```
+---------------------+
| Type = GL-TR-PR     | (1 byte)
+---------------------+
| Length              | (1 byte)
+---------------------+
| Maximum trees to compute | (2 byte)
```

Figure 6 Global Tree Proposal sub-TLV

Type: Router Capability sub-TLV type, TBD (GL-TR-PR)

Length: 2.

Maximum trees to compute: An unsigned 16-bit integer indicates maximum number of global trees the announcing RBridge wants area RB Bridges to calculate.
7.2.3. Global Tree Identifier sub-TLV

```
+----------------------+
| Type=GLTR-RT-IDs    | (1 byte)
|----------------------|
+----------------------+
| Length               | (1 byte)
+----------------------+
| Starting Tree Number | (2 bytes)
+----------------------+
| Nickname (K-th root) | (2 bytes)
+----------------------+
| Nickname (K+1 - th root) | (2 bytes)
+----------------------+
| Nickname (...)       | 
+----------------------+
```

Figure 7 Global Tree Identifier sub-TLV

Type: Router Capability sub-TLV type, set to TBD (GLTR-RT-IDs).

Length: 2 + 2*n, where n is the number of nicknames listed.

Starting Tree Number: This identifies the starting tree number of the nicknames that are trees for the domain. This is set to 1 for the sub-TLV containing the first list. Other Tree-Identifiers sub-TLVs will have the number of the starting list they contain. In the event a tree identifier can be computed from two such sub-TLVs and they are different, then it is assumed that this is a transient condition that will get cleared. During this transient time, such a tree SHOULD NOT be computed unless such computation is indicated by all relevant sub-TLVs present.

Nickname: The nickname at which a distribution tree is rooted.

7.3. Announcing Group Addresses

Group Address announcements facilitate optimization of multicast forwarding. [RFC6326] and [rfc6326bis], define series of sub-TLV to announce various flavors of Group addresses. These sub-TLVs are encapsulated in Group Address TLV (142). We propose to define new
set of sub-TLV under Group Address TLV to carry Group Address announcements applicable to Global trees. IS-IS Group Address TLV (142) does not have flags to control the scope of the TLV. Hence, explicit, sub-TLV definitions are required to indentify group announcements that have global scope.

New sub-TLV numbers are required for the following.

- Group MAC Address Sub-TLV
- Group IPv4 Address Sub-TLV
- Group IPv6 Address Sub-TLV
- Group Labeled MAC Address Sub-TLV
- Group Labeled IPv4 Address Sub-TLV
- Group Labeled IPv6 Address Sub-TLV

Above sub-TLVs as defined in [RFC6326] and [rfc6326bis] applies to all trees within the TRILL campus. New sub-TLV definitions include flexibility to define the applicable multicast trees. This flexibility allows applications to further optimize multicast pruning per multicast tree basis.

New group address sub-TLV will be named as below and they have common TLV header as defined in Figure 8.

- Group MAC Address-multicast tree Sub-TLV
- Group IPv4 Address-multicast tree Sub-TLV
- Group IPv6 Address-multicast tree Sub-TLV
- Group Labeled MAC Address-multicast tree Sub-TLV
- Group Labeled IPv4 Address-multicast tree Sub-TLV
- Group Labeled IPv6 Address-multicast tree Sub-TLV
Figure 8 Common structure of Group-Address-multicast tree sub-TLVs

- Type: G-ADDR-TR, (TBD). Defines sub-TLV for
  - Group MAC Address-multicast tree Sub-TLV
  - Group IPv4 Address-multicast tree Sub-TLV
  - Group IPv6 Address-multicast tree Sub-TLV
  - Group Labeled MAC Address-multicast tree Sub-TLV


- Topology ID: 2 byte identifier of the topology instance id.
8. Architecture Elements of Multi-level Multicast framework

- Bootstrap RBridge
- Rendezvous Point (RP)
- Default Affinity sub-TLV
- Area Affinity sub-TLV
- RP Election Protocol

Five main elements of the multi-level multicast framework are listed above. Functional overviews of the elements are discussed below. Details, such as state machines, PDU format, etc, will be presented in later versions of this document.

8.1. Bootstrap RBridge

Each Area has one more area border RBridges between the Area and the L2 backbone area. For each area one of its area border RBridges is elected as the Bootstrap RBridge.

Border RBridges communicate with each other using the TRILL BSR protocol. Each border RBridge has a configured priority to be a Bootstrap RBridge with system-ID as the tie breaker. The RBridge with the highest priority become the bootstrap RBridge for the area.
Bootstrap RBRidge selects the Rendezvous Point (RP) RBRidges and assign each RP a set of trees for which RP will function as the gateway between the local and global multicast trees. To avoid loops and/or packet duplication the set of trees MUST only be allocated to one and only one RP.

8.2. Rendezvous Point (RP)

Rendezvous Point (RP) RBRidge performs the function of gateway (or acts as a point of plumbing) between the local multicast tree and the corresponding global multicast tree rooted in the L2 backbone area.

Bootstrap RBRidge designates one of the border RBRidges (including itself) as the RP for a set of (mutually exclusive) trees.

Each border RBRidge, using TRILL BSR protocol, announces its desire to be an RP. The desire to be an RP is a configurable option and enabled by default to be an RP.

8.3. Default Affinity sub-TLV

Default Affinity sub-TLV is announced by each RP to inform the RBRidges in the L1 Area the association of the default nickname to a set of trees through the announcing RP [trillcmt].

RBRidges in the L1 area, based on the Default Affinity sub-TLV installs the RPF check for default nickname for the specified tree "t" on an interface facing towards the announcing RP.

8.4. Area Affinity sub-TLV

The Area Affinity sub-TLV announced by each RP to inform the RBRidges in L2 backbone Area the association of local L1 Area nicknames to a set of trees through the announcing RP (Figure 3).

RBRidges in the L2 backbone area or has interfaces to the L2 backbone area, based on the Area Affinity sub-TLV, installs the RPF check for the nicknames in the indicated L1 Area, for the specified tree "t", on an interface facing towards the announcing RP.

8.5. TRILL BSR Protocol

9. Security Considerations

TBD
10. IANA Considerations

IANA is requested to add the Area Affinity sub-TLV, Nickname Management sub-TLV, Global Tree capability sub-TLV, Global tree proposal sub-TLV, Global tree Identifier sub-TLV as sub-TLVs under Router capability TLV.

11. References

11.1. Normative References


11.2. Informative References


Senevirathne Expires August 21, 2012
12. Acknowledgments

We wish to thank Leonard Tracy, Dinesh Dutt and Ashok Ganesan for reviewing and providing constructive feedback.

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

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

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

Les Ginsberg  
CISCO Systems  
510 McCarthy Blvd.  
Milpitas CA 95035

Phone: 408-527-7729  
Email:ginsberg@cisco.com

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

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

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

Email: jon.hudson@gmail.com

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

Email: aldrin.ietf@gmail.com

Ayan Banerjee  
CISCO Systems  
425 East Tasman Drive  
San Jose CA 95134

Phone: 408-527-0539  
Email: ayabaner@cisco.com
ICMP based OAM Solution for TRILL
draft-tissa-trill-oam-03.txt

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on July 6, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.
This document presents a solution suite for TRILL data plane monitoring and failure detection. Methods presented herein allow non-cooperating IP payloads, exercising multi-paths, verifying multicast trees, locating end stations, virtual segments and diagnosing connectivity problems. ICMP protocol is proposed as framework for error reporting. Document also presents network wide health monitoring, distribution and reporting methods that are intended for efficient troubleshooting.

Table of Contents

1. Introduction...................................................4
   1.1. Motivation................................................6
   1.2. Contributors..............................................7
2. Conventions used in this document..............................7
3. Protocol Architecture Overview.................................7
   3.1. Overview of Tools.........................................8
   3.2. TRILL Data Plane..........................................9
   3.3. Monitoring...............................................10
   3.4. Traffic Triggered Monitoring (TTM).......................10
   3.5. Distribution.............................................10
   3.6. ISIS.....................................................11
   3.7. Reporting................................................11
4. Frame Format..................................................11
   4.1. Encoding of Request message..............................12
   4.2. Encoding of Response Message............................13
   4.3. Encoding of Notification Message........................13
      4.3.1. Pseudo IP Header..................................15
   4.4. OAM Command Messages....................................15
5. 127/8 in-band OAM IP address..................................16
   5.1. IPv6 default in-band address.............................16
6. Identification of Diagnostic frames............................17
6.1. Identification of Layer 2 Flow..........................17
6.2. Identification of IP Flows..............................17
6.3. Identification of Flows using Hop-Count Restrictions.....19
6.4. Identification of Multicast Flows.........................20
   6.4.1. Identification of overall tree verification frames..20
   6.4.2. Identification of Layer 2 Multicast group verification frames.....................................................21
   6.4.3. Identification of IP Multicast group verification frames........................................................................21
6.5. Default OAM flow Parameters..............................21
6.6. Validation of OAM Request and Response frames............22
7. ISIS Extensions...............................................23
8. ICMP multi part extensions....................................25
   8.1. ICMP Echo Request and Response message extensions...25
   8.2. C-Type Definitions.......................................26
9. Details of Diagnostic tools...................................57
   9.1. Loopback Message........................................57
      9.1.1. Theory of Operation..................................58
         9.1.1.1. Originator RBridge............................58
         9.1.1.2. Intermediate RBridge..........................59
         9.1.1.3. Destination RBridge............................59
      9.1.2. Loopback Message Hop-count method................60
         9.1.2.1. Identification of OAM frames....................60
      9.2. Prevent leaking out from TRILL network..............60
   9.3. Path Trace Message......................................61
      9.3.1. Theory of Operation..................................61
         9.3.1.1. Originator RBridge.............................61
         9.3.1.2. Intermediate RBridge...........................62
         9.3.1.3. Destination RBridge............................63
   9.4. Multicast Tree Verification (MTV) Message..............63
      9.4.1. Theory of Operation..................................64
         9.4.1.1. Originator RBridge............................64
         9.4.1.2. Intermediate RBridge..........................65
         9.4.1.3. In scope RBridges..............................66
   9.5. MAC address discovery Message..........................67
      9.5.1. Theory of Operation..................................68
         9.5.1.1. Originator RBridge.............................68
         9.5.1.2. Receiving RBridges.............................69
   9.6. Address-Binding Verification Message....................71
      9.6.1. Extension to ARP and invARP.........................72
         9.6.1.1. Encoding ARP-invARP extensions................74
   9.7. End-Station Attachment Point Discovery..................76
   9.8. DRB and AF Discovery....................................77
      9.8.1. Theory of Operation..................................78
         9.8.1.1. Originator RBridge.............................78
         9.8.1.2. Receiving RBridge.............................78
   9.9. Diagnostic Payload Discovery for ECMP coverage........80
1. Introduction

TRILL protocol has revolutionized how Layer 2 networks are being built and used. Legacy Ethernet networks provide single path for forwarding traffic and require all of the switches in the network to learn end-station MAC addresses. TRILL, on the other hand utilize multiple active links for forwarding thereby maximizing the overall network bandwidth utilization. TRILL is simple plug-and-play solution and does not require intermediate devices to learn MAC addresses of end-stations. These powerful characteristics of TRILL optimize performance and increase scaling limits. However, with that comes increased difficulty in diagnosing connectivity problems and locating end stations.
Network operators are used to troubleshooting legacy networks with single paths. Legacy devices maintain forwarding database of all end-station addresses in the Layer 2 network. Network administrators can trace the path taken by specific MAC address by examining the forwarding databases of devices. TRILL core switches, by design do not maintain end-station address database. Hence, administrators may not be able to trace a path taken by a specific MAC address by tracing the forwarding databases. Additionally, a given device may utilize multiple active paths to reach to a destination and may use a completely different forwarding topology for multicast traffic than it would use for unicast traffic. These challenges mandate the presence of an effective tool set to monitor and diagnose data plane failures in TRILL networks. These tools and protocols must stay as close as possible to the forwarding paths taken by actual data. OAM frames should not leak to end stations or out of the TRILL network to legacy networks.

TRILL base protocol specification [RFC6325] does not specify algorithm for selecting a path from a set of equal cost paths to forward a given flow. The majority of traffic in the networks is IP centric and most devices deploy some sort of hashing algorithm to identify the forwarding path from set of equal cost paths for a given flow. Thus, it is desirable to use IP address and TCP/UDP port information as inputs to the ECMP selection hash function. Use of such higher level information provides better distribution of flows across multiple equal cost paths. This document, propose a framework that allow specifying, various combinations of payloads including IP payloads and actual payloads.

As TRILL based networks get deployed, during the transition period, it may be required for TRILL R Bridges to co-exist with legacy networks. It is very helpful for the network operator if TRILL data plane failure detection tools allow isolating problem to specific legacy device or at least to the interface(s) that the downstream legacy device is connected. Solutions presented in this document facilitate identifying legacy devices or R Bridge interfaces legacy devices are connected to.

ICMP (Internet Control Message Protocol) [RFC 792] has been in use for nearly three decades. ICMP multipart extensions [RFC4884], propose methods to extend ICMP messages to include additional information, without changing or inventing new ICMP message types. In this document we utilize ICMP for reporting of errors. ICMP multipart extensions will be utilized to define additional information that is specific to TRILL. Additionally use of ICMP allows sending error reports either in-band or out-of-band. Use of out-of-band ICMP allows network operators to diagnose uni-
directional path failures easily. Also, the same ICMP infrastructure can be utilized to generate unsolicited error notifications for TRILL data plane failures, such as Destination unreachable, Time Exceed (TTL expiry), Parameter Mismatch (MTU mismatch) etc..

Availability of Network health information is a valuable starting point for any failure detection process. In this document we present the concept of network regions, monitoring of network regions and distribution of network health.

Diagnostic tools are also commonly referred to as OAM (Operations, Administration and Maintenance). In this document we use words diagnostics and OAM interchangeably. Unless explicitly specified both the words means the same.

1.1. Motivation

Currently published TRILL OAM solutions, [TRILLCH] and [TRILLOAM], mainly focus on data plane encoding and individual tools. The encoding methods presented in [TRILLCH] and [TRILLOAM], require defining OAM channel that utilize a special EtherType. Implementations that utilize ECMP selection algorithms based on higher layer address information may require flexible OAM channel that allow specifying different payloads including IP based payloads.

Availability of network health information is important for efficient isolation of network connectivity problems. Currently there are neither standard sets of such data to be distributed nor framework to distribute network health data. Lack of such leads to cumbersome and time consuming troubleshooting of network connectivity issues, especially in multi-vendor networks.

Device virtualization is an increasing trend in datacenters and large enterprises. Physical servers may host multiple virtual servers and these virtual servers may move from physical server to physical server based on load balancing policies. As part of network connectivity problem isolation, it is important to identify the location of the virtual servers and R Bridges they are connected to. Currently, administrators are required to utilize multiple tools to locate these virtual machines and connecting R Bridges.

ICMP has been in use over three decades as the primary OAM tool of IP infrastructure. It is highly desirable to utilize the framework of existing infrastructure such as ICMP, thereby leveraging knowledge, implementation and time to market.
TRILL networks can co-exist with multi access LAN networks at the boundary of the TRILL network. TRILL protocol [RFC6325], introduced Designated RBridge (DRB) and Appointed Forwarder (AF) concepts to ensure loop free forwarding and load splitting at the boundary of TRILL and multi access LAN networks. Discovery of DRB,AF and associated VLANs are important for effective fault isolation at the TRILL and multi access LAN boundary. Currently there are no known tools available for the purpose.

In this document we propose a framework and solution suite that will address the above.

1.2. Contributors

Many people contributed with ideas and comments. Among all, following people made notable contributions to all parts of this document and spend time reviewing, debating and commenting to ensure this specification address the problem space.

Ian Cox, Ronak Desai, Satya Dillikar , Rohit Watve, Ashok Ganesan and Leonard Tracy.

2. Conventions used in this document

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

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

3. Protocol Architecture Overview

Effective OAM solution is not only a set of tools but a wholesome solution that covers all aspects of OAM, such as tools, monitoring, reporting etc. Solution presented in this document contains multiple subcomponents that cover various elements of the total solution. There are six subcomponents in the proposed architecture. These subcomponents collectively are called TRILL OAM Protocol. Here we present an overview of the architecture of the solution and explain the purpose of each of subcomponents and interaction between different subcomponents. Subsequent sections cover details of each of the subcomponents.
3.1. Overview of Tools

The Tools subcomponent consists of series of utilities to implement various data plane monitoring and failure detections methods. Individual tools are invoked directly by the user or by the monitoring subcomponent. Individual tools allow, where applicable, for callers to specify options such as ECMP coverage, destination RBridge nickname, pay-load etc. Tools interface with the TRILL data plane layer to send and receive OAM frames. At the time of writing following tools are included as part of the tool set.

1. Loopback Message (Ping)
2. Path Trace Message (Trace route)
3. Multicast Tree Verification (mtv)
4. MAC discovery

5. Address Binding Verification

6. IP End-station Locator

7. DRB-AF discovery

8. Notification messages

9. OAM Command messages

Tools, based on the intended use, can be classified into 3 broader categories as below.

<table>
<thead>
<tr>
<th>Category</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault Verification</td>
<td>Loop Back Message</td>
</tr>
<tr>
<td>Fault Isolation</td>
<td>Path Trace Message, Multicast Tree Verification</td>
</tr>
<tr>
<td>Auxiliary</td>
<td>MAC discovery, Address Binding Verification, IP End-station Locator, DRB-AF Discovery, Error Notification, OAM command messages</td>
</tr>
</tbody>
</table>

3.2. TRILL Data Plane

The TRILL data plane receives and transmits frames on behalf of the tools subcomponent. As far as the encapsulation is concerned, TRILL data plane layer treat these frames exactly as it would treat a regular data frame. In fact one of the key design goals is to maintain TRILL data plane diagnostic (OAM) frames as close as possible to actual data frames. Additionally, implementation MUST satisfy the following requirements:

1. OAM frames SHOULD NOT leak in to legacy Ethernet or to end stations outside the TRILL cloud
2. RBridge MUST have ability to identify OAM (diagnostics) frames intended for a destination RBridge.

3. RBridgeS SHOULD have ability to identify TRILL data OAM frames that are not intended for itself and forward such frames without assistance from the CPU.

We explain in Section 6 various methods available to identify TRILL OAM (diagnostic) frames intended for the local RBridge and satisfy above requirements.

3.3. Monitoring

The Monitoring subcomponent utilize the tools subcomponent to monitor the TRILL data plane and proactively detect connectivity faults, configuration errors (cross connect errors) etc. The monitoring subcomponent provides options to specify frequency, retransmission count, ECMP choice and all other applicable options to the specific tool being used to implement the monitoring service. Based on the configuration specified by the user, the monitoring subcomponent periodically invokes the applicable tools. Additionally, based on configuration, monitoring results are propagated to the distribution subcomponent. Monitoring results are always associated with a monitoring region. The monitoring region is an administrative partition of the network such that it: 1. Maximize the fault coverage, 2. Optimize network health data summarization. More details of regions are discussed in Section 10.

The Monitoring subcomponent also interfaces with Traffic Triggered Monitoring subcomponent.

3.4. Traffic Triggered Monitoring (TTM)

Traffic Triggered Monitoring facilitates monitoring and diagnose of live data traffic. TTM subcomponent interfaces with the Data Plane to install required TTM policies. Details of the TTM framework and operations are presented in section 11.

3.5. Distribution

The distribution subcomponent has two primary inputs

- Data from the Monitoring Layer
- Data from other RBridges via ISIS GenApp

The distribution subcomponent performs the following functions:
o Advertising locally generated data
o Applying Advertising policies and re-advertising received data
o Maintaining the network health Database

Details of distribution layer and data handling are presented in section 10.

3.6. ISIS

TRILL OAM protocol suite proposed in this document utilize ISIS to distribute

OAM capability of individual RBridge

In-band OAM IP and MAC address

Above, OAM capability and In-band OAM address information are advertised using ISIS MT-Protocol extensions.[section 7.]

Network monitoring data are distributed using ISIS GenApp extension methods specified in [GenApp]. Details of encoding and proposed TLV definitions are defined in detail in section 7.

3.7. Reporting

The Reporting subcomponent allows users to define and use various reports on network health. The Reporting subcomponent utilize data available in the distribution subcomponent to generate requested reports. Sample reports are listed in Appendix A.

4. Frame Format

TRILL data plane diagnostic (OAM) frames can be broadly classified in to four types: request, response, notification and command messages. Request messages are generated to measure TRILL data plane characteristics, such as connectivity. Response messages are generated by a RBridge in response to a request. Notifications are unsolicited messages generated due to certain failures such as unreachable destination. OAM command messages provide a generic framework of communication between RBridges for OAM purposes. Details of individual messages are covered in later sections. Here we present frame encoding format for Request, Response and Notification messages.
4.1. Encoding of Request message

---

<table>
<thead>
<tr>
<th>Outer Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRILL Header</td>
</tr>
<tr>
<td>L2 Header + EthType ^</td>
</tr>
<tr>
<td>IP Header (including TCP/UDP)</td>
</tr>
<tr>
<td>User defined data or padded to zero</td>
</tr>
<tr>
<td>ICMP Header</td>
</tr>
<tr>
<td>Common ICMP Extension Header</td>
</tr>
<tr>
<td>ICMP extensions (optional)</td>
</tr>
</tbody>
</table>

Figure 2 Encoding TRILL data plane diagnostic request message

The above diagram depicts encapsulation of TRILL data plane diagnostic request frames. Encoded in the frame is the diagnostic payload. The diagnostic payload is a flexible structure that allows users to specify different kinds of payloads, including actual payloads. Most hardware implementations use IPDA:IPSA:DestPor:SrcPort based hash methods to select ECMP paths for IP frames. For non-IP payloads, R Bridges normally use a Layer 2 MAC DA and SA based hash for selecting an ECMP path. Flexible diagnostic payload allows users to drive end to end ECMP selection based on payload without needing additional hardware. Also, in terms of forwarding, this keeps diagnostic frames as close as possible to data frames. The length of the diagnostic payload must be deterministic. We propose a fixed 128 byte size for the diagnostic payload section of the OAM frame. This allows including IPv6 frames with multiple 802.1Q tags in to the diagnostic payload. The remaining bytes are set to zero, if the specified frame is smaller than the 128 byte fixed size.
ICMP header immediately follows the diagnostic payload. The ICMP header is constructed as defined in [RFC792] and [PINGEXT]. [PINGEXT] provide methods to extend ICMP echo request message to include ICMP multi part extensions.

ICMP multi part extensions [RFC 4884] are defined to carry additional information and are encoded after the ICMP header.[section 8.]

4.2. Encoding of Response Message

```
+-------------------------------+  Response Message
| TRILL Header or MAC Header   |
+-------------------------------+  \                                \ ^
| IP Header                    |  |                                |
+-------------------------------+  |                                |
| ICMP Header                   |  v                                |
+-------------------------------+  |                                |
| Common ICMP Extension Header  |  |                                |
+-------------------------------+  | original frame                 |
|                               |  . (TRILL Header +              |
|                               |  . diagnostic payload)          |
+-------------------------------+  |                                |
| ICMP extensions               |  v                                |
+-------------------------------+  |
```

Figure 3 Encoding of OAM response message

The above diagram depicts encoding of OAM response messages. If in-band delivery is requested, the OAM response message MUST be encoded as payload in a TRILL data frame. The ingress RBridge nickname MUST be set to the RBridge nickname of the node generating the response. Egress RBridge nickname MUST be set to the ingress RBridge nickname of the original TRILL data frame that triggered this response.

Normal IP forwarding rules MUST be followed, if an out-of-band response is requested.

4.3. Encoding of Notification Message

Notification messages are generated in response to an error condition such as delivery failure due to incompatible MTU or destination RBridge not in the forwarding table etc.. Out-of-band
responses are generally indicated by explicitly including the indication to receive an out-of-band response in the TRILL OAM request frame. Since notifications are generated proactively, the originator RBridge may not have methods to identify the IP address required to deliver an out-of-band response. Hence, in this document we propose to deliver Notification messages in-band. Delivery of out-of-band messages are outside the scope of this document.

The RBridge generating the Notification message MUST include up to 128 bytes of the original frame that triggered the notification message. If the original frame contains less than 128 bytes, then the remaining bytes MUST be padded with zeros.

```
+-------------------------------+
| TRILL Header                  |
+-------------------------------+
| IP Header                     |
+-------------------------------+
| ICMP Header                   |
+-------------------------------+
| Pseudo IP header             |
+-------------------------------+
| original frame               |
|   (TRILL Header + L2+ Ethtype |
|     + data)                  |
+-------------------------------+
| ICMP extensions              |
+-------------------------------+
```

Figure 4 Encoding of Notification message

The TRILL outer header of the frame that triggered the notification message is not included in the notification message. The Next-Hop header information in the original frame is of local significance to the specific link and may not be of interest to the originator of the data frame.

The Following error messages are currently supported

- Time Expired
- Destination Unreachable
- Parameter Problem
Additional TRILL OAM error codes may be specified as ICMP multipart extensions in above notifications messages. These error codes indicate the cause of the error. Please see section 8. for error code definitions and section 9.10. for theory of operation.

4.3.1. Pseudo IP Header

RFC 792 requires original payload section of ICMP messages, Time Expiry, Destination Unreachable and Parameter Problem to contain a valid IP header. RFC 1122 recommends ICMP implementations to multiplex incoming error notification messages to the related application based on the IP header information. The Pseudo IP header defined here intends to serve that purpose.

In this document we propose, for the purpose of TRILL OAM, to construct the pseudo IP header as a UDP header. IP addresses are derived based on the in-band IP addresses of the RBridges (section 5.). The destination port is the well known UDP destination port in the block of assigned "User Ports" (1024-49151). We intend to request IANA assignment of a UDP destination port for use in TRILL OAM.

4.4. OAM Command Messages

\[
\begin{array}{c}
\text{+-----------------------------+} \\
\text{| TRILL Header or MAC Header |} \\
\text{+-----------------------------+} \\
\text{| IP Header | ^} \\
\text{+-----------------------------+} \\
\text{| ICMP Header |} \\
\text{+-----------------------------+} \\
\text{| Common ICMP Extension Header | Command Message} \\
\text{+-----------------------------+} \\
\text{| ICMP extensions |} \\
\text{+-----------------------------+} \\
\end{array}
\]

Figure 5 Encoding of OAM Command Message

OAM command messages are originated by RBridges to indicate other RBridges in the network to execute commands on behalf of the originating RBridge. OAM command messages are not required to follow a specific ECMP path. Hence, OAM messages do not contain a diagnostic payload section.

Destination IP address of the OAM command message is either in-band OAM IP address or out-of-band management IP address of the
destination RBridge. Incoming OAM command message are delivered to the ICMP stack by the IP stack. ICMP stack further identify the message as an OAM message due to embedded ICMP extensions. ICMP stack delivers OAM command message to the OAM processing module for further processing.

The TTM (Traffic Triggered Monitoring) framework presented in section 11. and the Diagnostic Payload discovery presented in section 9.9. extensively utilizes OAM command messages.

5. 127/8 in-band OAM IP address

In this document we propose to use same ICMP framework deployed in IP infrastructure for communicating OAM information. RBrigdes are not required to have IP interfaces enabled. However, in order to receive and process ICMP messages, RBrigdes are required to have at least a pseudo IP address. In this document, we propose to use 127/8 addressing scheme similar to the MPLS data plane failure detection methods [RFC 4379]. It is important that each RBridge have a straightforward method of identifying corresponding in-band OAM IP address of any given RBridge, without additional processing or lookups.

The 127/8 Address range is allocated for internal loopback addresses [RFC 1122] and required not to be routed. RFC 4379 updates RFC 1122 to utilize 127/8 addressing to communicate between devices in a peer-to-peer model that does not require routing. In this document, we propose to use 127/8 addressing model to identify in-band IP address required for OAM purposes. Additional methods are provided as ISIS LSP extension to announce, other addresses, user may desire to use for OAM in-band purpose. By default all RBrigdes MUST support the 127/8 addressing model specified here.

Each RBridge nickname is 16bits wide [RFC6325]. Let’s assume RBridge nickname RB is divided in RB(msb) and RB(lsb), such that, RB(msb) takes the upper 8bits of the RB and RB(lsb) takes the lower 8bits of the RB. Corresponding in-band IP address of RB is 127.RB(msb).RB(lsb).100. Implementation MUST facilitate methods to avoid conflicts between in-band OAM address and implementation specific 127/8 address allocations.

5.1. IPv6 default in-band address

IPv6 based systems have two options to derive the in-band IP address. The systems may choose, IPv6 native loopback address
::RBid:100 or IPv4 mapped IPv6 addressing format of ::FFFF:127.RB(msb).RB(lsb).100.

RFC 4379, MPLS Data Plane failure detection methods, utilize IPv4 mapped IPv6 addressing. One of the design objectives of the proposal is to re-use as many existing OAM extensions as possible. Hence, implementation that support IPv6 MUST utilize the IPv4 mapped IPv6 addressing format for default IPv6 in-band address. Deployments that desire to utilize native addressing MAY advertise native IPv6 in-band address using OAM extensions in section 7.

6. Identification of Diagnostic frames

In this document we have proposed to use the TRILL header as defined in [RFC6325], without modifications. The standard TRILL header currently, does not provide option to identify diagnostic frames. Hence, it is important to have circumstantial methods to identify diagnostic frames intended for the local RBridge and prevent leaking of diagnostic frames outside of TRILL network. In this section we explain, various methods to attain the above goals.

6.1. Identification of Layer 2 Flow

As stated earlier, most RBridges use Destination and Source MAC address, combination to determine the next hop ECMP interface to forward non IP frames. It is required to provide flexibility for the user to specify destination MAC address and source MAC address. We propose to use special EthType (TBD) to identify OAM (diagnostic) frames that contain non IP diagnostic payloads.

Each RBridge, if TRILL data plane OAM enabled, MUST provide following processing:

- Forward frames that have egress RBridge nickname equal to local RBridge nickname and EthType equal to Diagnostic Ethtype, to the Central Processing Unit (CPU). Such frames SHOULD NOT egress out of the RBridge.
- The RBridge SHOULD not egress frames with Diagnostic Ethtype to non TRILL interfaces.

6.2. Identification of IP Flows

As stated earlier, most RBridges use combination of IP address and Layer 4 information such as UDP/TCP Port, to determine the next hop ECMP interface to forward IP frames. Hence, it is important to provide flexibility for users to specify destination IP addressing and payload information.
In this section we propose several approaches to identify OAM (diagnostic) frames with IP payloads that are addressed to the local RBridge for processing

Method 1:

Use of Well known Destination MAC address:

We propose to use a well known diagnostic MAC address (TBD-DMAC-1), as the Destination MAC address of the inner Layer 2 header.

Each RBridge, if TRILL data plane diagnostic is enabled, MUST provide the following processing:

- Forward frames which have egress RBridge nickname equal to the local RBridge nickname and Destination MAC address of the inner Layer 2 header equal to the Well Known Diagnostic MAC address (TBD-DMAC-1) to the Central Processing Unit (CPU). If RBridge nickname is not equal to the local RBridge nickname, frame MUST be forwarded normally.
- RBridge SHOULD NOT egress frames with the Diagnostic MAC address (TBD-DMAC-1) as destination address to non TRILL interfaces.

Method 2:

Use of Well known Source MAC address:

We propose to use a well known source MAC address (TBD-SMAC-1), as the source MAC address of the inner Layer 2 header.

Each RBridge, if TRILL data plane diagnostic is enabled, MUST provide following processing:

- Forward frames that have egress RBridge nickname equal to the local RBridge nickname and source MAC address of the inner Layer 2 header equal to Well Known source MAC address (TBD-SMAC-1), to the Central Processing Unit (CPU). If the egress RBridge nickname is not equal to the local RBridge nickname then the frame MUST be forwarded normally.
- Each RBridge SHOULD NOT egress frames with Well known MAC address as source address to non TRILL interfaces.
- RBridge SHOULD NOT dynamically learn the well known Source MAC address (TBD-SMAC-1) specified above.
Method 3:

Use of RBridge specific OAM MAC address:

Each RBridge may advertise, MAC address for the purpose of receiving OAM frames with IP payloads. Sending RBridges may use the advertised MAC address as the destination MAC address of the inner Layer 2 header of originating diagnostic request frames.

Each RBridge, if TRILL OAM is enabled MUST provide following processing:

- Forward frames that has egress RBridge equal to the local RBridge nickname AND Destination MAC address of the inner Layer 2 header equal to the advertised RBridge specific OAM MAC address, to the Central Processing Unit (CPU).
- RBridge SHOULD NOT egress frames with RBridge specific OAM MAC address as destination address to non TRILL interfaces.

6.3. Identification of Flows using Hop-Count Restrictions

Methods presented in Sections 6.1. and 6.2. utilize one or more fields in the data frame to identify OAM frames against real data frames. As a result, operator does not have complete flexibility of specifying all of the fields in the diagnostic payload. This restriction while acceptable in most cases may not be acceptable in some cases. There may be instances that operator desire to specify the exact frame under investigation.

RFC 6325 section 3.6 explains handling of TRILL Hop-Count field. Accordingly, frames received with Hop-Count of zero (0) MUST not be forwarded.

OAM frames that wishes to utilize Hop-Count restriction process MUST first discover the Hop-Count from ingress RBridge to the egress RBridge. Hop-Count discovery may be accomplished using Path Trace message specified in section 9.3.

Desired OAM frame is then encoded using methods specified in this document. Hop-Count field of the TRILL header is updated with the above discovered Hop-Count value.

Additionally, it is recommended, to invalidate the inner diagnostic payload IP checksum, if the specified diagnostic payload is an IP packet. Invalidation of the inner diagnostic payload IP checksum prevent end stations processing of OAM packets, in the unlikely event of such OAM packets leaking out to of the TRILL network.
Egress RBridge processing routines MUST have methods to identifying OAM frames with Hop-Count expiry from actual data frames with Hop-Count expiry. OAM frame validation process specified in section 6.6. MUST be followed. A frame MUST be treated as a data frame with Hop-Count expiry, if the OAM validation process specified in section 6.6. failed.

6.4. Identification of Multicast Flows

Multicast frames are forwarded using one of the available multicast trees in the TRILL network [RFC6325]. Selection of a multicast tree is done at the ingress RBridge. Multicast frames are directed to a selected multicast tree at the ingress. Hence exact payload definition is not required for the purpose of ECMP selection. However, based on multicast pruning, certain multicast addresses may not be required to be forwarded to all members of the tree. Intermediate switches perform, (S,G) or (*,G), forwarding based on IP addresses for IP frames and MAC address for non IP frames. Hence, in order to verify the effect of multicast pruning users may require methods to specify Layer 2 and/or IP addressing information, as applicable. There are two types of multicast tree verification messages:

- Overall Tree Verification Messages
- Pruned Tree Verification Messages

6.4.1. Identification of overall tree verification frames

We propose to utilize a well known multicast diagnostic MAC address (TBD-GMAC-1) for this purpose. If TRILL data plane diagnostics are enabled, this specific MAC address MUST be installed on every RBridge for all tress and MUST NOT be subject to pruning.

Each RBridge performs (*,G) forwarding of the frames based on the well known multicast diagnostic MAC address (TBD-GMAC-1) in the inner Layer 2 destination address. Additionally, it sends a copy of the frame to the CPU for analysis and generates a response to the requester. Please see section 8.3 for details of multicast tree verification message processing.

A RBridge SHOULD NOT egress multicast frames with above diagnostic MAC address in to non TRILL interfaces. Also, RBridge MUST discard any native frame received on non TRILL interfaces with the above diagnostic MAC address as the destination MAC address.
6.4.2. Identification of Layer 2 Multicast group verification frames

We propose to utilize the diagnostic EthType (TBD) that was defined earlier for identification of Layer 2 group verification frames. User SHOULD have the ability specify destination MAC address, source MAC Address, VLAN and payload data up to 128 octets.

Each RBridge, performs standard multicast forwarding. Additionally, if EthType of the frame is equal to the well known diagnostic Ethtype (TBD), the RBridge sends a copy of the frame to the CPU for analysis and generating response to the requester. Please see section 9.3 for details of multicast tree verification message processing.

RBridge MUST NOT egress multicast frames with above EthType in to non TRILL interfaces. Also, RBridge MUST discard any native frame received on non TRILL interfaces with the above EthType.

6.4.3. Identification of IP Multicast group verification frames

We propose to use the well known MAC address (TBD-SMAC-1) defined in section 6.2 as the source MAC address. Users have flexibility to define, IP Address, VLAN and other payload data upto 128 octets. The Destination MAC address is derived based on the IP Multicast destination address.

RBrigdes perform (S,G) or (*,G) forwarding using the IP address information. Additionally, each RBridge send a copy of the frame to the CPU, if the source MAC address matches the well known MAC address defined here in.

RBridge MUST NOT egress multicast frames with above source MAC address to non TRILL interfaces. Also, each RBridge MUST discard any native frame received on a non TRILL interfaces with the above source MAC address.

RBridge MUST NOT dynamically learn the well known source MAC address specified here.

6.5. Default OAM flow Parameters

Parameters specified herein SHOULD be utilized as default parameters. Parameters specified under the Fixed category MUST not be changed based on user specification and MUST be followed exactly as specified below.
<table>
<thead>
<tr>
<th>Flow type</th>
<th>Default Values</th>
<th>Fixed fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 2</td>
<td>DA= Well Known MAC</td>
<td>EthType=OAM(TBD)</td>
</tr>
<tr>
<td></td>
<td>SA= RBridge Interface MAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VLAN=native VLAN</td>
<td></td>
</tr>
<tr>
<td>IPv4 OR</td>
<td>IP Address = in-band address</td>
<td>EthType=0x8000 OR</td>
</tr>
<tr>
<td>IPv6</td>
<td>IP Dest. Port = 3503</td>
<td>EthType=0x86DD</td>
</tr>
<tr>
<td></td>
<td>IP Src. Port = TBD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DA = OAM MAC of egress RBridge</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SA = ingress RBr interface MAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VLAN=native VLAN</td>
<td></td>
</tr>
<tr>
<td>Multicast Tree Verification</td>
<td>SA= RBridge Interface MAC</td>
<td>DA=Well Known Multicast MAC EthType=OAM(TBD)</td>
</tr>
<tr>
<td></td>
<td>VLAN=native VLAN</td>
<td></td>
</tr>
<tr>
<td>Layer 2 Multicast</td>
<td>DA= Well Known MAC</td>
<td>EthType=OAM(TBD)</td>
</tr>
<tr>
<td></td>
<td>SA= RBridge Interface MAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VLAN=native VLAN</td>
<td></td>
</tr>
<tr>
<td>IP Multicast</td>
<td>IP Dest Address = Default OAM MCast address</td>
<td>EthType=0x8000 OR</td>
</tr>
<tr>
<td></td>
<td>IP Src. Address = in-band-address</td>
<td>EthType=0x86DD</td>
</tr>
<tr>
<td></td>
<td>IP Dest. Port = 3503</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IP Src. Port = TBD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DA = OAM MAC of egress RBridge</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SA = ingress RBr interface MAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VLAN=native VLAN</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6 Default Parameters of Diagnostic(OAM) Payloads

6.6. Validation of OAM Request and Response frames

OAM processing module MUST further validate the received request/response messages to ensure their compliance to this specification using the methods specified herein.
OAM messages are encoded as specified above and contain an ICMP Header and an ICMP Common Header as specified in [PINGEXT].

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+-------------------------------+
|Version|        Length         |           Checksum            |
+-------------------------------+-------------------------------+
|                  Magic-Number (0x54726163)                     |
+-------------------------------+-------------------------------+
```

Figure 7 ICMP Common Extension Header

The OAM process MUST offset to the common header and validate the Version and Magic-number fields. The Version MUST be one (1) and Magic-number MUST be 0x54726163. If the Version or the Magic-number does not match, then the frame is not an OAM frame.

If these fields are matching the specified values, then the checksum is calculated over the Version, Length and Magic number fields. The calculated checksum is compared against the checksum in the frame. If the two values do not match then the frame is not an OAM frame.

Frames that pass both the tests above are further qualified as below.

The Length field in the common ICMP header specifies the starting location of the ICMP Extension. The first ICMP Extension is the Version and Flags Extension (C-type 1) (Section 8.1).

Version and Flag fields of c-type 1 MUST be validated to identify whether the OAM frame is of a known version. OAM frames of unknown versions are discarded.

Frames that pass all of the above tests are valid OAM frames and further processed according to the OAM code specified in the Version and Flags Extensions.

7. ISIS Extensions

A new ISIS subTLV definition is required to announce the following OAM related information:

- OAM capability
- OAM in-band IP address
- OAM in-band MAC address
We propose to define a single sub TLV structure within ROUTER-CAPABILITY ISIS TLV (242), to announce the above OAM information.

```
+--------+
| Type   |
+--------+
| Length |
+--------+--------+
| ver | Res | v | i | m | o |
+-------------------+
| Sender nickname |
+-------------------+
| OAM MAC address |
+-------------------+
| OAM in-band IP address |
+-------------------+
```

Figure 8 ISIS extension for OAM

Type : (1 octet) TBD (one of the sub-TLV definitions under MT-PORT-CAP ISIS TLV)

Length : (1 octet) Length of the subTLV, in octets, excluding Type and Length fields. Minimum 2.

Ver : (4 bits) indicate the OAM version. Currently set to zero.

Res : (1 octet), Reserved for future use. Set to zero on transmission and ignored on receipt.

V : (1 bit) if set, indicates IP address included in the TLV is IPv6. Only one of I or V bit MUST be set. If both are set, it is a malformed TLV and must be discarded without further processing.

I : (1 bit) if set indicate IP address included in the TLV is IPv4. Only one of I or V bits MUST be set. If both are set, it is malformed TLV and must be discarded without any further processing.

M : (1 bit) If set, indicates MAC address is included in the TLV.

O : (1 bit) If set, indicates announcing RBridge is OAM capable.
MAC Address : (6 octets), IEEE MAC address, associated with the in-band IP address. If included, the MAC address MUST precede the IP address.

IP Address : (4 or 16 octets), OAM in-band IP address. If present MUST follow MAC address.

Above PDU encoding MUST follow exact order as specified and fields are not interchangeable.

NOTE: Both I and V flags MAY be set to zero to indicate that announcing RBRidge desire to use the default OAM address. The default OAM address is the 127/8 address derived as specified in section 5.

8. ICMP multi part extensions

We propose to utilize a new Class-Num [RFC4884] to identify TRILL OAM related extensions specified in this document and other related documents. IANA has established a registry for ICMP extensions and we intend to seek a Class-Num assigned for this purpose.

Within the TRILL OAM Class-Num, C-Types are defined and registered in the IANA to identify various different extensions specified herein and other related future documents.

8.1. ICMP Echo Request and Response message extensions

RFC 4884 proposes a framework to extend ICMP message types: Time Expiry, Parameter Problem and Destination Unreachable. RFC 4884 therefore cannot be applied to extend other ICMP messages, such as ICMP echo request and response messages. ICMP Echo request and response is by far the most widely used OAM tool. Extensibility of ICMP Echo request in a backward compatible manner is very important. Such a framework provides flexibility to the ICMP message structure to carry application specific information.

[PINGEXT] presents a framework to extend ICMP messages in a backward compatible manner and allow encoding specific extensions in RFC 4884 compliant c-types.

In this document, we propose to utilize the framework presented in [PINGEXT] to extend the ICMP echo request or response structures encoded within the TRILL OAM messages.
8.2. C-Type Definitions

C-Types defined in this section MUST be embedded in the ICMP Extension object format proposed in section 8 of RFC 4884. Figure 9 presents the format of the ICMP Extension object defined in RFC 4884. Figure 9 is entirely for reference purposes only and readers are referred to RFC 4884 for most up to date information.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +---------------------------------------------+
  | Length | Class-Num | C-Type |
  +---------------------------------------------+
  +---------------------------------------------+
  // (Object payload) //
  +---------------------------------------------+
```

Figure 9 ICMP Extension Object

Section below defines the format of the object payloads, only. ICMP Object header MUST precedes object payloads defined in section 8.2. Figure 10 below presents an example of encoding C-Type 1, i.e Version and Flags object.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +---------------------------------------------+
  | Length | Class-Num | C-Type |
  +---------------------------------------------+
  | Version | code | Reserved | F|c|o| |
  +---------------------------------------------+
```

Figure 10 Example of Encoding Version and Flags object

Version and Flags: C-Type 1

Contain Version number, code and associated flags. Currently Out-of band Request, Final and Cross Connect Error flags are defined.
<table>
<thead>
<tr>
<th>Bits</th>
<th>31</th>
<th>24</th>
<th>16</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
</table>
|      | +------------+---------+-----------+--+--+--+
|      | | Version | code | Reserved | F | C | o |
|      | +------------+---------+-----------+--+--+--+

Figure 11  C-Type 1, Version and Flags

Version (8 bits): Currently set to zero

Code (1 octet): TRILL OAM Message codes. See below for currently available TRILL OAM Message codes.

Reserved (22 bits): Set to zero on transmission and ignored on receipt

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

C (1 bit): Cross connect error (VLAN mapping error), if set indicates VLAN cross connect error detected. 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.

TRILL OAM Message codes:

0 : Loopback Message Request
1 : Loopback Message Response
2 : Path Trace Request
3 : Path Trace Response
4 : Time Expiry Notification (error)
5 : Parameter Problem Notification (error)
6 : Destination Unreachable (error)
7 : Multicast Tree Verification Request
8 : Multicast Tree Verification Response
9 : MAC Address discovery Request
10 : MAC Address discovery Response
11 : DRB discovery request
12 : DRB discovery response
13 : AF discovery request
14 : AF discovery response
15 : AF-VLAN discovery request
16 : AF-VLAN discovery response
17 : TTM Set Message
18 : TTM Get Message
19 : TTM Remove Message
20 : TTM Response Message
21 : TTM Indication Message
22 : Payload Generation request Message
23 : Payload Generation response Message
24 : Loopback Message request with Hop-count
26 - 255 : Reserved

Originator IP Address: (C-type 2)

Length of the ICMP extension header indicates whether the address is IPv4 or IPv6. Please refer to RFC 4884 for ICMP extension encoding and ICMP header structure.

```
Bits
  31 0
+-----------------------------+
 |                           |
 . IP Address               |
 +-----------------------------+

Figure 12  C-Type 2 Originator IP address
```

Upstream Identification: (C-type 3)

The Upstream Identification C-type structure encodes upstream path information such as upstream neighbor nickname, ingress interface index (ifindex) and name of the ingress port.
Bits

+------------------------+---------------+
|  nickname             | Reserved      |
|                        |               |
|                         | ifindex       |
|                         |               |
|                         | Slot          |
|                         | Port          |
|                         | Speed         |
|                         | State         |

Figure 13  C-Type 3 Upstream Identification

Nickname (2 octets): TRILL 16 bit nickname of the upstream RBRIDGE. [RFCtrill]

Reserved (2 octets) : Reserved, set to zero on transmission and ignored on receipt.

Ifindex (2 octets) : unsigned integer of local significance

Slot (2 octets) : Slot number

Port (2 octets) : Port number

Speed (2 octets) : Speed in 100Mbps. Zero (0) indicates port speeds less than 100Mbps.

State (2 octets) : Represent the state of the port.

0: Down - no errors
1: Disable
2: Forwarding-no errors
3: Down - errors
4: Forwarding - errors
5: Forwarding - oversubscribed
6: Link Monitoring disable
All other values reserved.

Monitored VLAN(diagnostic VLAN ) : (C-type 4)

Monitored VLAN c-type include in the ICMP extensions allows for testing the integrity of the inner payload VLAN and the expected VLAN. The expected VLAN is encoded in the Monitored VLAN c-type. The destination RBRIDGE, compare the VLAN of the inner payload with the VLAN value encoded in the Monitored VLAN c-type. If these two VLAN
values mismatch, RBridge SHOULD set the cross connect flag in the response. A RBridge MUST NOT set the cross connect error flag for other than the above specified VLAN mismatch scenario.

<table>
<thead>
<tr>
<th>Bits</th>
<th>16</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>VLAN</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-----</td>
<td></td>
</tr>
</tbody>
</table>

Figure 14  C-Type 4 Monitored (Diagnostic) VLAN

Downstream Identification: (C-Type 5)

The Downstream Identification C-type carries multiple sets of data, each corresponding to individual downstream neighbor among collection of equal cost paths.

<table>
<thead>
<tr>
<th>Bits</th>
<th>31</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecmp count</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>nickname</td>
<td></td>
</tr>
<tr>
<td>ifindex</td>
<td>Next hop neighbor information</td>
<td></td>
</tr>
<tr>
<td>Slot</td>
<td>Port</td>
<td></td>
</tr>
<tr>
<td>Speed</td>
<td>State</td>
<td></td>
</tr>
<tr>
<td>Repeat next hop neighbor identification for each neighbor</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 15  C-Type 5 Downstream Identification

Ecmp count (2 octets): Number of equal cost paths to the given destination from this RBridge.

Reserved (4 octets): Reserved, set to zero on transmission and ignored on receipt.

Next-hop neighbor information:
Nickname (16 bits): TRILL 16 bit nickname [RFCtrill]

Ifindex (2 octets): unsigned integer of local significance

Slot (2 octets): Slot number

Port (2 octets): Port number

Speed (2 octets): Speed in 100Mbps. Zero (0) indicates port speeds less than 100Mbps.

State (2 octets): Represent the state of the port.
0: Down - no errors
1: Disable
2: Forwarding-no errors
3: Down - errors
4: Forwarding - errors
5: Forwarding - oversubscribed
6: Link monitoring disable
All other values reserved.

NOTE: Repeat Next-hop neighbor identification entry per each ECMP.
Total number of neighbor entries MUST equal to ecmp count.
Individual neighbor entry MAY have variable length.

Path for this payload: (c-Type 6)

Path for this payload indicates the next hop neighbor that this frame could have been forwarded on based on the payload hashing.
Bits

<table>
<thead>
<tr>
<th>31</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>nickname</td>
<td>Reserved</td>
</tr>
<tr>
<td>ifindex</td>
<td></td>
</tr>
<tr>
<td>Slot</td>
<td>Port</td>
</tr>
<tr>
<td>Speed</td>
<td>State</td>
</tr>
</tbody>
</table>

Figure 16  C-Type 6 Path for this payload

Nickname (16 bits): TRILL 16 bit nickname [RFCtrill]

Ifindex  (2 octets) : unsigned integer of local significance. 0xFFFF indicate CPU.

Slot     (2 octets) : Slot number

Port     (2 octets) : Port number

Speed    (2 octets) : Speed in 100Mbps. Zero (0) indicates port speeds less than 100Mbps.

State    (2 octets) : Represent the state of the port.

0: Down - no errors
1: Disable
2: Forwarding-no errors
3: Down - errors
4: Forwarding - errors
5: Forwarding - oversubscribed
6: Link monitoring disable
All other values reserved.

DRB Information (c-Type 7)

| 31 | 16 | 8 | 0 |
|------------------|---------------|
| nickname         | state | R|P|

Figure 17  Nickname of the DRB

Senevirathne Expires July 6, 2012
Nickname (2 octets) : TRILL nickname of the DRB

State (1 octet) : DRB state

R (7 bits) : Set to zero on Transmission and ignored on receipt

P (1 bit) : Set when pseudo node bypass is indicated by the DRB for the link

AF Information (C-Type 7)

Follow the same encoding as C-Type 6, above.

Nickname and state are of the AF.

Enable VLAN List (c-Type 8)

```
  31  27  16  12  0
+------------------------+
| R | St-VLAN | R | End-VLAN |
+------------------------+
```

Figure 18  Enabled VLAN List

R (4 bits) : Reserved, set to zero on transmission and ignored on receipt.

St-VLAN (12 bits) : Start VLAN

End-VLAN (12 bits) : End VLAN

Start VLAN and End VLAN represent the range of enabled VLANS. If the VLAN range is non contiguous, then multiple Enabled VLAN lists MUST be included, each representing a contiguous VLAN set.

Announcing VLAN set (c-Type 9)

Announcing VLAN list uses the same format as the Enable VLAN List (c-Type 8)
R (4 bits) : Reserved, set to zero on transmission and ignored on receipt.

St-VLAN (12 bits) : Start VLAN

End-VLAN (12 bits) : End VLAN

Start VLAN and End VLAN represent the range of announcing VLANS. If the VLAN range is non contiguous, then multiple of announcing VLAN list MUST be included, each representing a contiguous VLAN set.

AF List (c-Type 10)

This c-Type lists the VLANS for which responding RBridge is a the appointed forwarder.

Reserved (2 octets) : set to zero on transmission and ignored on receipt.

Nickname (2 octets) : TRILL 16 bit nickname of the RBridge

R (4 bits) : Reserved, set to zero on transmission and ignored on receipt.

St-VLAN (12 bits) : Start VLAN

End-VLAN (12 bits) : End VLAN
AF List MUST be repeated for each of the contiguous VLAN ranges that the responding RBridge function as Appointed Forwarder.

**DRB Life Time (c-Type 11)**

DRB Life time indicates the Life time, of the DRB operational role, of the RBridge.

```
31                                     0
+--------------------------------------+
|                                      |0
+   Life Time                          +
|                                      |1
+--------------------------------------+
```

**Figure 21** DRB Life Time

Life Time (8 octets): Indicates the Life time of the operational role in seconds.

**AF Lifetime (C-Type 12)**

AF Life time indicates the Life time, of the AF operational role, of the RBridge for the specified VLAN.

Encoding follow the same format specified in C-Type 11.

**Designated VLAN changes (C-Type 13)**

Indicates number of times a given RBridge has observed Designated VLAN changes. Each change may potentially lead to traffic disruptions.

```
15            0
+-------------+
| Change count|
+-------------+
```

**Figure 22** Number of times Designated VLAN changes

Change count (2 octets): Indicates number of times a given RBridge has observed Designated VLAN changes

**RBridge scope List (c-Type 14)**
15           0
+-----------+
|  R  |  Nu |
+-----------+
|   nickname 1 |
+-----------+
.           .
|   nickname n |
+-----------+

Figure 23   Scope List c-Type 14

R (1 octet) : Reserved, zero on transmission and ignored on receipt.

Nu (1 octet) : number of nicknames listed

Nickname 1 .. n (2 octets) each: List TRILL RBridge nickname of in scope RBriges.

Nicknames MUST be numerically sorted. With nickname1 the lowest to nickname n the highest. This facilitate easy processing the receiving RBridge.

Nu = 0 indicate no embedded nicknames in the message and response required from all RBriges, where applicable.

Multicast Tree downstream List (c-Type 15)
Multicast Tree downstream list provides information on downstream leaf Rbridges on the specified tree.

<table>
<thead>
<tr>
<th>Bits</th>
<th>31</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>leaf count</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td>nickname</td>
</tr>
<tr>
<td></td>
<td>ifindex</td>
<td>Downstream leaf information</td>
</tr>
<tr>
<td></td>
<td>Slot</td>
<td>Port</td>
</tr>
<tr>
<td></td>
<td>Speed</td>
<td>State</td>
</tr>
<tr>
<td></td>
<td>Repeat downstream leaf information for each downstream RBridge</td>
<td></td>
</tr>
</tbody>
</table>

Figure 24  C-Type 5 Multicast Tree Downstream List

Leaf count (16 bits): Number of R Bridges downstream to this R Bridge.

Downstream leaf information:

Nickname (16 bits): TRILL 16 bit nickname [RFC3312]

Ifindex (32 bits):Unsigned 32 bit integer that has only a local significance to the sending R Bridge. Value 0xFFFF indicates CPU interface.

Slot (2 octets): Slot number
Port (2 octets): Port number
Speed (2 octets): Speed in 100Mbps. Zero (0) indicates port speeds less than 100Mbps.
State (2 octets): Represent the state of the port.
0: Down - no errors
1: Disable
2: Forwarding-no errors
3: Down - errors
4: Forwarding - errors
5: Forwarding - oversubscribed
All other values reserved.

NOTE: Repeat downstream RBridges reachability information per each leaf node. Total number of neighbor entries MUST equal to leaf count. Individual neighbor entry MAY have variable length.

MAC-discovery Address List (c-Type 16)

```
+------------+
| count      |
+------------+
| MAC        |
| Address 1  |
+            |
+            |
+            |
| Address n  |
+------------+
```

Figure 25  MAC-discovery Address List

Count (2 octets) : Number of MAC addresses embedded in the response
MAC Address ( 6 octets) : 6 octet MAC address

MAC-discovery response Address List (c-Type 17)
Figure 26   MAC-discovery response

Count (2 octets) : Number of MAC addresses embedded in the response

T     (4 bits ) : Type of MAC address 0 – Dynamic, 1 Static, 2-15 Reserved
VLAN (12 bits): VLAN identifier associated with the MAC address

L (8 bits): Length of Service Tag in bits.

Service Tag (4 octets): Service Tag is right aligned. For 24 bit Length, left most 8 bits of Service Tag MUST be set to zero.

MAC Address (6 octets): 6 octet MAC address

Age (8 octets): Age of the MAC address in seconds. For a static MAC address, this field is ignored.

Ifindex (4 octets): Interface index on which MAC address is learnt

Slot (2 octets): Slot number of the interface on which this MAC address is learnt

Port (2 octets): Port number of the interface on which this MAC address is learnt.

vNTAG (2 octets): virtual TAG identifier associated with the MAC address. Value 0 indicate no vNTAG association with the MAC address.

Speed (2 octets): Speed in 100Mbps. Zero (0) indicates port speeds less than 100Mbps.

State (2 octets): Represent the state of the port.

0: Down - no errors
1: Disable
2: Forwarding-no errors
3: Down - errors
4: Forwarding - errors
5: Forwarding - oversubscribed
6: Un-monitored
All other values reserved.

Error code (c-Type 18)

Error code c-Type allows an RBridge to specify various error codes within high-level notification messages such as Time Expiry, Parameter Problem and Destination unreachable. The sub-error codes within each of the error code allow specifying further details of the error.
Error Code (2 octets) : Identify the error. Currently following errors are defined

0 - VLAN non existent
1 - VLAN in suspended state
2 - Cross connect error
3 - Unknown RBridge nickname
4 - Not AF
5 - MTU mismatch
6 - Interface not in forwarding state
7 - Service Tag non existent
8 - Service Tag in suspended state
9 - 0xFFFF - Reserved for future use and MUST not be used in transmission.

Sub-code (2 octets) : identify the sub-error code.
0 - 0xFFFF - Reserved for future use and MUST not be used in transmission.

Warning code (c-Type 19)

Warning code c-Type allow a RBridge to specify various error codes within high-level notification messages such as Time Expiry, Parameter Problem and Destination unreachable. The sub-warning codes within each of the warning codes allow to specify further details of the warning.

Warning Code (2 octets) : Identify the Warning. Currently following Warnings are defined
0 - Invalid RBridge nickname (RBridge nickname in the range 0xffco to 0xffff)
1 - Invalid VLAN (Reserved VLAN)
2 - AF VLAN list Mismatch
3 - 0xFFFF - Reserved for future use and MUST not be used in transmission.

Sub-code (2 octets) : identify the sub-error code.
0 - 0xFFFF - Reserved for future use and MUST not be used in transmission.

Information code (c-Type 20)

Information code c-Type allow a RBridge to specify various information codes within the high-level notification messages such as Time Expiry, Parameter Problem and Destination unreachable. The sub-info codes within each of the code allow specifying further details of the information.

```
+------------------+--------------+
| Information Code | sub-code     |
+------------------+--------------+
```


Figure 29   C-Type 20 Information code

Information Code (2 octets) : Identify the Information. Currently following Information are defined

0 - 0xFFFF - Reserved for future use and MUST not be used in transmission.

Sub-code (2 octets) : identify the sub-error code.
0 - 0xFFFF - Reserved for future use and MUST not be used in transmission.

Diagnostic-Payload (c-Type 21)

The Disagnostic-Payload c-Type encodes Trill-header and diagnostic payload for response messages or original frame for notification messages. The length of the embedded diagnostic-payload is indicated by the Length in the C-type header ([RFC4884]).
Diagnostic-Payload : 0 or more 32bit words.

This c-type MUST only be included in Response or notification messages only. It MUST only occur exactly once within the message.

Data (c-Type 22)

The Data c-Type facilitates encoding of any arbitrary set of data in the OAM messages. Such Opaque data may be utilized to generate TRILL OAM frames with different lengths. It may also be utilized for other purposes, such usage methods are outside the scope of this document.

Data-Payload : 0 or more 32bit words.

This c-type may occur zero or more times within a given OAM message.

Service Tag (c-Type 23)

Overlay Technologies such as [FNGRAIN], utilize Identification Tags that are wider than the 12bit VLAN Tag used in IEEE 802.1Q. Objective of these tags, regardless of the width, is to identify virtual service instance within the overlay network. Hence, in this document the tag is referred to as Service Tag.
Service Tag: 4-octets wide opaque value.

Applications that requires 24bit service Tags MUST set upper 8bits to zero in transmission and discards requests received with non zero value in upper 8bits.

Control Plane Forwarding Verification Request (c-Type 24)

Downstream Identification (c-Type 5) presented earlier facilitate users to discover forwarding paths available on the dataplane to reach the specified destination. It is often desirable to discover control and data plane inconsistencies. Control Plane Forwarding Verification c-Type facilitate the users to optionally obtain Forwarding information available on the control plane.

Ecmp Count : (2 octets) : Ecmp Identifier indicates the ECMP to verified. Value 0xFF indicate all of the ECMP needed to be verified.

Egress nickname : (2 octets), nickname of the destination RBridge.

Control Plane Forwarding Verification Response (c-Type 25)

Control Plane Forwarding Verification Response is generated in response to c-Type 24 above.
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ECMP count</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ECMP Identifier</td>
<td>nickname</td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ifindex</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Slot</td>
<td>Port</td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Speed</td>
<td>State</td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Repeat next hop neighbor information for each neighbor.

Figure 34  C-Type 25 control Plane Forwarding Verification Response

Ecmp count (2 octets): Number of equal cost paths to the given destination from this RBridge.

Reserved (2 octets): Reserved, set to zero on transmission and ignored on receipt.

Next-hop neighbor information:

ECMP Identifier: ECMP Identifier for this record.

Nickname (2 octets): TRILL 2 octet nickname [RFCtrill]. Value 0xFFFF indicates requested ECMP Identifier is invalid.

Ifindex (2 octets): unsigned integer of local significance

Slot (2 octets): Slot number

Port (2 octets): Port number

Speed (2 octets): Speed in 100Mbps. Zero (0) indicates port speeds less than 100Mbps.

State (2 octets): Represent the state of the port.

0: Down - no errors
1: Disable
2: Forwarding-no errors
3: Down - errors
4: Forwarding - errors
5: Forwarding - oversubscribed
6: Link monitoring disable
All other values reserved.

NOTE: Repeat Next-hop neighbor identification entry per each ECMP. Total number of neighbor entries MUST equal to ecmp count. Individual neighbor entry MAY have variable length.

Reverse Path Forwarding Verification Request (c-Type 26)

Downstream Identification (c-Type 5) presented earlier facilitate users to discover forwarding paths available on the data plane to reach the specified destination. It is often desirable to discover the reachability and ECMP information along the reverse path. C-Type presented here allows users to discover Reverse Path to a specified RBRidge from the receiver. This is an optional parameter and can be included in Loopback messages, Path Trace messages and OAM Command messages.

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ecmp Identifier</td>
</tr>
<tr>
<td>nickname</td>
</tr>
</tbody>
</table>

Figure 35 C-Type 26 Reverse Path Forwarding Verification Request

Ecmp Count : (2 octets) : Ecmp Identifier indicates the interested Reverse Path ECMP. Value 0xFF indicate all of the ECMP.

nickname : (2 octets), nickname of the destination RBRidge.

Reverse Path Forwarding Verification Response (c-Type 27)

Reverse Path Forwarding Verification Response is generated in response to c-Type 26 above.
### Ecmp count (2 octets): Number of equal cost paths to the given destination from this RBridge.

Reserved (2 octets): Reserved, set to zero on transmission and ignored on receipt.

Next-hop neighbor information:

<table>
<thead>
<tr>
<th>ECMP Identifier:</th>
<th>ECMP Identifier of this record.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nickname (2 octets):</td>
<td>TRILL 2 octet nickname [RFCtrill]. Value 0xFFFF indicates requested ECMP Identifier is invalid.</td>
</tr>
<tr>
<td>Ifindex (2 octets):</td>
<td>unsigned integer of local significance</td>
</tr>
<tr>
<td>Slot (2 octets):</td>
<td>Slot number</td>
</tr>
<tr>
<td>Port (2 octets):</td>
<td>Port number</td>
</tr>
<tr>
<td>Speed (2 octets):</td>
<td>Speed in 100Mbps. Zero (0) indicates port speeds less than 100Mbps.</td>
</tr>
<tr>
<td>State (2 octets):</td>
<td>Represent the state of the port.</td>
</tr>
</tbody>
</table>

0: Down - no errors
1: Disable
2: Forwarding-no errors
3: Down - errors
4: Forwarding - errors
5: Forwarding - oversubscribed
6: Link monitoring disable
All other values reserved.

NOTE: Repeat Next-hope neighbor identification entry per each ECMP. Total number of neighbor entries MUST equal to ecmp count. Individual neighbor entry MAY have variable length.

Traffic Triggered Monitoring (TTM) Profile (c-Type 28)

Details of Traffic Triggered Monitoring are presented in section 11. TTM profile defines the container c-Type for the TTM profile. With the TTM profile c-type, other related c-types are included. Included c-types are linked through next c-type field. Value zero in next c-type field indicate end of included c-types.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Priority                     |         Identifier            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| C |  F  |            Frequency                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Count                                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Reserved                |      Next c-type              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 37   C-Type 28 TTM Profile

C Indicate the Class

TTM Profile action (c-Type 29)

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Action       | Reserved    |         Next c-type             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 38   C-Type 29 TTM action

Action (2 octets):
0: count RX packets
1: Count TX packets
2: Count RX bytes
3: Count TX bytes
4: Log
5: Capture
6: - 0xFF reserved

NOTE: Given TTM Profile may contain multiple actions. E.g. count TX, count RX and Log.

TTM Test Point (TP) (c-type 30)

Figure 39   C-Type 30 TTM Test Point

NOTE: Given TTM Profile may contain multiple Test Points.

TTM Ingress End Point (c-type 31)

Figure 40   C-Type 31 TTM Ingress End Point

T 3 bits:
1: TRILL RBridge nickname (Length of End Point is 2 octets)
2: IPv4 End Point (Length of the End Point is 4 octets)
3: IPv6 End Point (Length of the End Point is 16 octets)
4: 7 Reserved.

End Point Address: Address of the End Point as defined by the T value.

Next c-type is the c-type of the next information. Value zero indicates this as the last c-type.

TTM Egress End Point (c-type 32)

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| T   |  Reserved               |    next c-type                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|   End Point Address                                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 41   C-Type 32 TTM Egress End Point

T 3 bits:

5: TRILL RBridge nickname (Length of End Point is 2 octets)
6: IPv4 End Point (Length of the End Point is 4 octets)
7: IPv6 End Point (Length of the End Point is 16 octets)
8: 7 Reserved.

End Point Address: Address of the End Point as defined by the T value.

Next c-type is the c-type of the next information. Value zero indicates this as the last c-type.
### C-Type 33 TTM Pattern

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>Reserved</td>
<td>next c-type</td>
<td></td>
</tr>
</tbody>
</table>

- **TTM Pattern**
- **TTM Pattern mask**

Figure 42 C-Type 33 TTM Pattern

**T 4 bits:**

- 0: TRILL ingress RBridge nickname (Length of the pattern is 2 octets)
- 1: TRILL egress RBridge nickname (Length of the pattern is 2 octets)
- 2: IPv4 Source End Point (Length of the pattern is 4 octets)
- 3: IPv4 Destination End Point (Length of the pattern is 4 octets)
- 4: IPv6 Source End Point (Length of the pattern is 16 octets)
- 5: IPv6 Destination End Point (Length of the pattern is 16 octets)
- 6: Source MAC address (Length of the pattern is 6 octets)
- 7: Destination MAC address (Length of the pattern is 6 octets)
- 8: EthType (Length of the pattern is 2 octets)
- 9: VLAN (Length of the pattern is 2 octets) Right justified, upper 4 bits are don't care.
- 10: Service Tag 24 bits. Right aligned with upper octet do not care.
- 11: Service Tag 32 bits
  - All other values Reserved.

**TTM Pattern Mask** defines the mask of the specified pattern. Length of the pattern mask is identical to the length of the address.

**Next c-type** is the c-type of the next information. Value zero indicates this as the last c-type.

**TTM Opaque Pattern** (c-Type 34)
Figure 43  C-Type 34 TTM Opaque pattern

Length: (1 octets) define the length of the TTM pattern in octets.

Offset: (1 octets) defines the offset from the pre-amble of the frame the specified pattern MUST be applied.

TTM Pattern is the pattern to be matched. Length of the pattern is specified by the Length field.

TTM Pattern mask is the mask for the specified pattern. Length of the pattern is specified by the Length field.

NOTE: Only one TTM Opaque pattern MUST be included in a given TTM profile. TTM profiles with more than one Opaque Pattern MUST be rejected.

End Point (c-type 35)

End Point c-type (35) indicate the address on the device that is generating the message. For TRILL this represent the 16 bit nickname of the RBridge.
T 3 bits:

9: TRILL RBridge nickname (Length of End Point is 2 octets)
10: IPv4 End Point (Length of the End Point is 4 octets)
11: IPv6 End Point (Length of the End Point is 16 octets)
12: 7 Reserved.

End Point Address: Address of the End Point as defined by the T value.

TTM Test Payload (c-type 36)

TTM Profile allow users to inject test frames from an intermediate device. C-type 35 End Point allows specifying egress end point of the tunnel or RBridge. C-type 36 presented here provide methods of specifying the required frame.
Test frame is the payload of the frame, excluding pre-amble and FCS.

Seed Destination MAC address (c-type 37)

Seed Destination MAC address is used when discovering diagnostic payloads combinations that span certain ECMP path combination. A given payload discovery request may contain multiple Seed MAC addresses. The identification field within the seed MAC address uniquely identifies a specific seed. MAC address field within the seed is divided into 6 fields. Each of these fields is named MA-1 to MA-6 and one octet wide. MA-x can take any legal value specified by IEEE MAC address specification. Non zero value in MA-x indicates that specific octets cannot be changed by the downstream RBridge when generating the diagnostic payload. MA-x fields with zero indicates either it is a fixed field or field that is available for downstream Rbridges to derive appropriate payload value. Each of the Max with zero value has corresponding C-type 39, MAC-Octet bit vector. Each bit in the MAC-Octets Mask indicates a valid value for that MA-x field. MAC-Octet Mask of zero length indicates the corresponding MA-x field has fixed value zero.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1  
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
|       Identifier      |Reserve|    MAC-0      |  MAC-1        |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| MAC-2         |    MAC-3      |    MAC-4      |  MAC-5        |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
```

Figure 46  C-Type 37 Seed Destination MAC address

Identifier (12 bits) Uniquely identify a MAC address seed within a diagnostic payload discovery message.

MAC-0 to MAC-5 Represent an octet in the IEEE MAC address and may take any legal value specified in IEEE 802.1. Any MAC-x field of value zero MUST have a corresponding C-type 39, MAC-Octet bit vector.

Seed Source MAC address (c-type 38)

Seed Source MAC address, c-type 38, has same format as c-type 37.

MAC-Octet bit vector (c-type 39)
Identifier (12 bits), uniquely identifies a MAC address seed within a diagnostic payload discovery message.

MAC-x : Value 0-5 indicates the MAC address octet represented by this bit-vector.

Bit-offset (octet) indicates the starting value of the bit-0 of bit vector. E.g. when bit-offset is 40, starting value of bit-0 is 40, bit-1 is 41 and so on.

Length (octet) indicates the length of the bit vector in bits.

A value 1 in a bit vector position indicates the value represented by that bit is an applicable value to be considered for the MAC-x field.

Payload generation request (c-Type 40)

ECMP start, ECMP end : (1 octet each) : ECMP start and ECMP end indicate the ECMP to verified. Value 0xFF in ECMP start and ECMP end indicate all of the available ECMP needed to be verified.

Egress nickname : (2 octets), nickname of the destination RBridge.

Payload generation response (c-Type 41)
Figure 49  C-Type 41 Payload generation response

ECMP Identifier: (1 octet):

Egress nickname : (2 octets), nickname of the destination RBridge.

Res : (6 bits), set to zero on transmission and ignored on receipt.

S : (3 bits) indicates the status.

  0. Success
  1. ECMP does not exist
  2. Unable to generate payload using the proposed seed
  3. System overloaded try later
  4. - 7 Reserved MUST not be used.

TTM command Response sub-codes (c-Type 42)

Figure 50  C-Type 42 TTM command Response sub-code

Sub-codes (1 octet):

  0 : Set response
  1 : Get Response
  2 : Remove Response
  3- 255 : are reserved and must not be used.

Reserved (1 octet): set to zero on transmission and ignored on receipt.

Status (1 octet):

  0 : Success
  1 : TTM profile does not exist
  2 : Remove failed
3 : Get failed
4 : Set failed - resource exceeded
5 : Set failed - other reasons

6-255 : Reserved and MUST not be used

EthType (c-Type 43)

<table>
<thead>
<tr>
<th>Reserved</th>
<th>Eth Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 51  C-Type 43 Eth Type

Eth Type (2 octets): Represent IEEE Ether Type

Reserved (2 octets): Set to zero on transmission and ignored on receipt.

9. Details of Diagnostic tools

In this section we present details of various diagnostic tools that are identified as part of the solution. We assume, readers are familiar with frame encoding methods, diagnostic frame identification methods, and ISIS and ICMP extensions presented earlier in the document. In this section we will only make reference to the extensions and methods, please refer to prior section for details.

9.1. Loopback Message

Loopback message is utilized for fault verification. It verifies connectivity between two RBridges, for a specified flow. Monitoring subsystem may use Loopback Message for connectivity monitoring and proactive fault detection. Users may specify exact flow, part of it or not at all. Additionally, users may also specify, ECMP choice at the ingress. ECMP choice can be a specific index, set of index, all of the index or non. If no ECMP index specified, payload is used to determine the ECMP choice. Method of deriving the ECMP choice using payload is implementation dependent and is outside the scope of this document. However, CPU generating the Loopback message SHOULD use the same ECMP selection algorithm as the data plane. Additionally some implementation may allow users to specify the ingress interface
that actual flow may ingress to the RBridge. Although ability to inject the data plane diagnostic frames from the ingress interface is optional feature, it is highly desirable, as it allows verifying end-end connectivity from an ingress port to an egress RBridge.

Egress RBridge can send its response either in-band or out-of-band. In-band-response, additionally allow to measure round trip delay. In-band responses are tagged with the same VLAN as the request frame. ICMP multi part extensions in the request message allow user to specify whether out-of-band response required. If out-of-band request required, IP address it desire to receive the response MUST be specified.

Additionally, diagnostic VLAN, may be specified as part of the ICMP multi part extensions. Receiver RBridge may compare inner VLAN in the payload and the specified diagnostic VLAN. If the two specified VLAN values do not match, C flag in Version C-type SHOULD be set to indicate cross connect error.

9.1.1. Theory of Operation

9.1.1.1. Originator RBridge

Identify the destination RBridge based on user specification or based on location of the specified address (see below sections for MAC discovery and address locator).

Construct the diagnostic payload based on user specified parameters. Default parameters MUST be utilized for unspecified payload parameters. See Figure 6 for default parameters.

Construct the ICMP Echo request header. Assign applicable identification number and sequence number for the request.

ICMP multi part extension Version MUST be included and set appropriate flags. Specify the code as Loopback Message Request(0).

Construct following ICMP multipart extensions, where applicable

- Out-of-band response request
- Out-of-band IP address
- Diagnostic VLAN
Specify the Hop count of the TRILL data frame per user specification. Or utilize the applicable Hop count value, if TRILL TTL is not being specified.

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

RBridge may continue to retransmit, the request at periodic interval, until a response received or re-transmission count expires. At each new re-transmission sequence number may be incremented.

9.1.1.2. Intermediate RBridge

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

9.1.1.3. Destination RBridge

Destination RBridge performs, frame identification methods specified in above section 5. If the Loopback message is addressed to the local RBridge, then the RBridge forward the Loopback messages to the CPU for processing. CPU performs frame validation and constructs the response as stated below.

Construct the IP header for the ICMP echo response. If no out-of-band response requested, IP address in the IP header MUST be in-band IP address. If out-of-band response requested destination IP address is the IP address specified in the request message. Source IP address is derived based on the outgoing IP interface address.

Construct the ICMP echo reply header using the received ICMP echo request.

Include the received TRILL header and diagnostic payload in to the data field of the ICMP echo request frame [section 4.2.].

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.

Error handling:
If VLAN cross connect error detected or inner.VLAN does not exist in the RBridge then generate Destination Unreachable message and specify the cause using error codes.

9.2. Loopback Message Hop-count method

The Loopback message procedures presented in section 9.1. utilize customers specified payload to derive the diagnostic payload embedded in the OAM message. Encoding methods presented in section 6. require that certain fields of the diagnostic payload to contain some fixed well-known values. Time to time operators may desire to include identical payload fields with no modifications. Hop-count method presented in this section facilitates inclusion of un-modified payload. When unmodified payloads are included as the diagnostic payload, there MUST be methods to identify such OAM frames from regular data frames and there MUST be methods to prevent such OAM frames leaking out of TRILL network.

9.2.1. Identification of OAM frames

Egress RBridge receives loopback messages employing hop-count method as hop-count expired frames. There MUST be methods to identify OAM frames employing hop-count expiry method from other frames that experience hop count expiry.

Firstly, procedures specified in section 6.6. MUST be utilized by the egress RBridge to differentiate receiving hop-count expired OAM frames from data frames.

Secondly, the egress RBridge identifies the hop-count expired OAM messages from loopback messages utilizing hop-count expiry method by examining OAM Message code. OAM messages utilizing hop-count expiry method MUST specify TRILL OAM Message code as "Loopback Message request with Hop-count" (24).

9.2.2. Prevent leaking out from TRILL network

First, the ingress RBridge that is generating the loopback message MUST discover the TRILL hop count to the egress RBridge. Hop count to the egress RBridge MAY be discovered either using the Path Trace Message specified in section 9.3. or some other method. The discovered Hop count MUST be used as the hop count included in the TRILL header.

Further, if the specified payload is IP, the IP header checksum SHOULD BE invalidated. The invalidation of IP checksum, prevents end
stations further processing the OAM frames, in the unlike event it reached the end station.

All other operations are similar to Loopback Message processing presented in section 9.1.

9.3. Path Trace Message

Primary use of Path Trace Message, commonly known in the IP world as "traceroute", is fault isolation. It may also be used for plotting path taken from a given RBridge to another RBridge. Operation of Path Trace message is identical to Loopback message except, that it is first transmitted with a TRILL Hop count field value of 1. Sending RBridge expect a Time Expiry message from the next hop or a successful response. If a Time Expiry message is received as the response, the originator RBridge record the information received from intermediate node that generated the Time Expiry message and resend 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 reach a configured maximum value.

9.3.1. Theory of Operation

9.3.1.1. Originator RBridge

Identify the destination RBridge based on user specification or based on location of the specified address (see below sections for MAC discovery and address locator).

Construct the diagnostic payload based on user specified parameters. Default parameters MUST be utilized for unspecified payload parameters. See Figure 4 for default parameters.

Construct the ICMP Echo request header. Assign applicable identification number and sequence number for the request.

ICMP multi part extension Version MUST be included and set appropriate flags. Set the code to Path Trace Request (2)

Construct following ICMP multipart extensions, where applicable

- Out-of-band response request
- Out-of-band IP address
- Diagnostic VLAN
Specify the Hope Count of the TRILL data frame as 1 for the first frame. Or use Hope Count value incremented by 1 if this is a retransmission generated in response to received Time Expiry message.

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

RBridge may continue to retransmit, the request at periodic interval, until a response received or re-transmission count expires. At each new re-transmission sequence number may be incremented.

9.3.1.2. Intermediate RBridge

Intermediate RBridge receive the diagnostic frame as Hope count expired frame. Based on flow encoding methods explained in above section 5, RBridge identify TRILL data plane diagnostic frames from actual data frames with Hope count expiry. Hop count time expiry messages may be generated for actual data frames as well. However, Hop count expiry message for actual data frames are always sent in-band, as actual payload does not have methods to specify the response delivery method.

CPU of intermediate RBridge that receives OAM frame with Hope count expiry performs following:

Identify whether in-band or out of band response requested. Construct the IP header accordingly.

Construct the ICMP Time Expiry message as specified in RFC 792 and RFC 4884. RFC 4884 specifies format of ICMP header when including ICMP multipart messages.

Include original TRILL header and diagnostic payload of the original frame as data for ICMP Time Expiry message. Update the length field to reflect the size of the TRILL header and diagnostic payload.

Include following ICMP multipart extensions

Version

Set the code to Path Trace Response (3)

Nickname of the RBridge
Information of the ingress interface (speed, state, slot, port)

Index of the interface where frame was received

Nickname of the upstream RBridge the frame was received

Downstream ecmp count

List of Downstream RBridges (nickname, interface index and interface information)

Downstream path this specific payload take (RBridge nickname, interface index and interface information)

Optionally include following ICMP multipart extensions

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

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

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

9.3.1.3. Destination RBridge

Processing is identical to section 8.1.1.3

9.4. Multicast Tree Verification (MTV) Message

Multicast Tree Verification messages allow verifying multicast tree integrity and Multicast address pruning. IGMP snooping is widely deployed in Layer 2 networks for restricting forwarding of multicast traffic to unwanted destinations. This is accomplished by pruning the multicast tree such that for specified (S,G,VLAN) or (*,G,VLAN), only required destinations are included in the outgoing interface list. It is possible due to timing or implementation defects, inaccurate pruning of multicast tree, may occur. Such events lead to incorrect multicast connectivity. Multicast tree verification and Multicast group verification messages are design to detect such multicast connectivity defects. Additionally, these tools can be used for plotting a given multicast tree within the TRILL network.

Multicast tree verification OAM frames are copied to the CPU of every intermediate RBridge that are part of the Multicast tree being
verified. Originator of the Multicast Tree verification message, specify the scope of RBridges that a response is required. Only, the RBridges listed in the scope field response to the request. Other RBridges silently discard the request. Definition of scope parameter is required to prevent receiving large number of responses. Typical scenario of multicast tree verification or group verification involves verifying multicast connectivity to selected set of end-nodes as opposed to the entire network. Availability of the scope, facilitate narrowing down the focus only to the interested RBridges.

Implementations MAY choose to rate limit CPU bound multicast traffic. As result of rate limiting or due to other congestion conditions, time to time, MTV messages may be discarded by the intermediate RBridges and requester may be required to retransmit the request. Implementations SHOULD narrow the embedded scope of retransmission request only to RBridges that has failed to respond.

9.4.1. Theory of Operation

9.4.1.1. Originator RBridge

User is required at minimum to specify either the multicast trees that needed to be verified or Multicast MAC address and VLAN or VLAN and Multicast destination IP address. Alternatively, for more specific multicast flow verification, user MAY specify more information e.g. source MAC address, VLAN, Destination and Source IP addresses. Implementation, at minimum, must allow user to specify, choice of multicast trees, Destination Multicast MAC address and VLAN that needed to be verified. Although, it is not mandatory, it is highly desired to provide option to specify the scope.

Default parameters MUST be used for unspecified parameters. Please refer to Figure 6 for default payload parameters for MTV message.

Based on user specified parameters, originating RBridge identify the nickname that represent the multicast tree.

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

Construct the diagnostic payload based on user specified parameters. For overall multicast tree verification message only multicast tree is specified as input. For generic multicast group verification, additional information such as group address is specified. Based on user provided parameters, implementation SHOULD identify whether the request is for overall multicast tree verification or for specific group verification.
For overall multicast tree verification, use well known multicast destination MAC address (TBD_GMAC-1) defined in above section 6.4.1. as the inner destination MAC address of the TRILL frame. Remaining parameters are derived based on default values specified in Figure 6

Construct ICMP echo request message header and include sequence number and identifier. Identifier and sequence number facilitate the originator to map the response to the correct request.

Version ICMP multipart extension MUST be included.

Code MUST be specified as Multicast Tree Verification Request (7)

Optionally, include following ICMP multipart extensions, where applicable

- Out-of-band response request
- Out-of-band IP address
- Diagnostic VLAN
- In scope RBridge list.
  - NOTE: "Nu" field in ICMP extension RBridge scope (section 8.2.) MUST be set to zero, if response required from all RBridges.

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

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

RBridge may continue to retransmit, the request at a periodic interval, until a response received or re-transmission count expires. At each new re-transmission sequence number may be incremented. At each re-transmission, RBridge may further reduce the scope to the RBridges it has not received a response.

9.4.1.2. Intermediate RBridge

Intermediate RBridges identify multicast verification frames per the procedure explained in section 6.4.
CPU of the RBridge validate the frame and analyze the scope RBridge list. If 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, RBridge proceed to 9.3.1.3 for further processing.

9.4.1.3. In scope RBridges

RBridge go through following processing, upon identifying that it’s nickname is specified in the scope RBridge list.

Identify whether in-band or out-of-band response requested.
Construct the IP header accordingly.

Construct the ICMP echo response message as specified in RFC 792.

Include TRILL header and diagnostic payload of the received OAM message as data of the ICMP response message.

Include following ICMP multipart extensions
Version, update the code as Multicast Tree Verification Response (8)
Nickname of the RBridge
Name of the ingress interface frame was received
Interface index where frame was received
Nickname of the upstream RBridge the frame was received
Downstream leaf node count
Leaf RBridge list {RBridge nickname, interface index and interface name}

Optionally, if VLAN cross connect error detected, then set C flag (cross connect error) in the versions extension.

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

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

RBridge MUST generate applicable notification messages if any error such as inner VLAN not available, detected against the OAM message.

9.5. MAC address discovery Message

MAC address discovery message is defined to discover following information

- RBridge nickname where the MAC address is learnt
- Interface Index and Name on which the MAC address is learnt
- Type (i.e. Static, Dynamic, Secure etc.)
- Age of the MAC address
- Virtual Interface Tag (vNTAG)
- Interface Type (Legacy or TRILL Shared)
- DRB on the VLAN (If Applicable)
- AF for the VLAN (If Applicable)
- Time AF operational (If Applicable)

Optionally, an implementation may include the following information

- System MAC address of the device connected to the port with which the MAC address is associated.

- System information, such as name, IP address and location of the device connected to the port with which the MAC address is associated.

- Information related to this MAC address from the remote device.

The method of obtaining the above optional information is outside the scope of this document. However, implementation may consider link level control protocols such as LLDP for the purpose.
9.5.1. Theory of Operation

There are two possible options to implement MAC address discovery. Either we may define a new MAC-discovery ISIS sub-TLV and use ESADI to propagate the request (similar to the MAC-Reachability TLV [RFC6165]) OR we may use multicast tree verification message and include a ICMP multipart extension to indicate that the message is a MAC discovery message.

Using the ISIS based method has disadvantage of being non real time and subjected to protocol delays. The second method above is independent of any control plane protocol implementation and can be exercised in real-time. Hence, in this document, we propose to utilize second method.

9.5.1.1. Originator RBridge

Use the well known Multicast MAC address described in section 6.4.1. above as the inner destination MAC address of the diagnostic payload. Use the applicable source MAC address and VLAN. Use the diagnostic EthType defined earlier as the EthType. Pad the remainder of the diagnostic payload with zero.

Construct ICMP echo request message and include sequence number and identifier. The sequence number and identifier facilitate the originator to map the response to the correct request.

Construct following ICMP multipart extensions

- Version
- Set the OAM code to the MAC address discovery request (9)
- Indicate that this is a MAC discovery message
- One or more MAC address to be discovered
- VLAN ID of MAC addresses (optional)
- Service Tag that represent the overlay network (optional)
- Out-of-band response request (optional)
- Out-of-band IP address (optional)
- In scope RBridge list. If response required from all RBridges, then the Nu count in RBridge scope list MUST be set to zero.
Specify the TTL value of the TRILL data frame to the applicable value.

Set the egress RBridge nickname to the nickname of the multicast tree used for broadcast and unknown unicast.

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

An RBridge may continue to retransmit the request at periodic interval until re-transmission count expires. At each new re-transmission sequence number may be incremented. The RBridge scope list of re-transmission messages MUST be pruned to include only the response pending RBridges. It is possible that more than one RBridge has learnt the requested MAC address. Hence the implementation MUST wait until the total wait time expires and SHOULD NOT abort the discovery process on receiving a single response.

9.5.1.2. Receiving RBridges

CPU of Intermediate RBridges receives a copy of the MAC discovery frame through methods explained in section 6.4.2. and 6.4.1.

Receiving in scope RBridges analyze the embedded ICMP multipart extensions to identify whether the request is for MAC discovery.

If the request is for MAC discovery, then the receiving RBridge queries its forwarding database to identify, whether requested MAC address is present with specified VLAN information.

The receiving RBridge generate responses only for identified MAC entries. If there are no matching MAC entries, the receiving RBridge silently discards the MAC discovery request.

If a matching MAC address is found, the receiving RBridge generates a Destination unreachable ICMP message (Type = 3) and code = 12, "Destination host unreachable for type of service". This essentially indicates, it has found the MAC address but has reached the end of the TRILL network where the MAC address is located.

RFC 4884 allow extension of ICMP messages. Only ICMP messages Destination Unreachable, Time Expired and Parameter Problem are currently extensible in RFC 4884 compliant manner. Other messages are only extensible for known payload size and considered non compliant to RFC 4884. For MAC discovery messages there is no
requirement to include original data payload. Also response to MAC
discovery can contain large amount of MAC address information.
Hence, we conclude to utilize Destination unreachable message as
opposed to using an ICMP echo response with fixed pay load size.

The receiving RBridge constructs the response as follows:

Construct the IP header based on the requested response type, in-
band or out-of-band. For an in-band response, use RBridge in-band IP
address. For an out-of-band response, use the provided egress
RBridge out-of-band address.

Construct the ICMP Destination Unreachable message per section 4.1
of RFC 4884. Specify, ICMP type=3 and code = 12. Specify the length
as zero. (i.e, no data included and ICMP extensions directly
follow).

Construct the pseudo IP header per section 4.3.1.

Include the following ICMP multi part extensions;

   nickname of this RBridge. (This is required in the event of out -
of band response to identify the originating RBridge nickname)

      Version

      Code, set to MAC address discovery response (10)

Additionally, include the following ICMP multipart extensions, for
each MAC address that was specified in the request and is present in
the RBridge forwarding DB:

   o Interface Index and Interface Information
     (Speed,Slot,Port,State) on which MAC address learnt

   o Type (i.e. static, Dynamic, Secure etc.)

   o Age of the MAC address

   o Virtual Interface Identification (vNTAG)

   o Interface Type (Legacy or Trill Shared)

   o DRB on the VLAN (If Applicable)

   o AF for the VLAN (If Applicable)
Optionally an implementation may include the following information:

- The system MAC address of the device connected to the port with which the MAC address is associated.
- System information, such as name, IP address and location of the device connected to the port on which MAC address is associated.
- Information related to this MAC address from the remote device.

If the response size is greater than the maximum MTU size of the outgoing interface, then multiple responses MAY be generated. The final response frame MUST contain ICMP multipart extension Version (C-Type 1) with F (final response) flag set.

The response frame is delivered to the TRILL data plane for in-band response.

If out of band response was requested, the response frame is delivered to the IP protocol stack.

9.6. Address-Binding Verification Message

Virtual machine provisioning is a very common practice in data centers and enterprises. It is normal for virtual machines to move from one physical machine to another physical machine. As a result ARP tables on gateways can be stale and network operators may need to resort to multiple tools to identify the location of a given IP address that is being diagnosed for connectivity. Even if the location of the server that host the given IP address is identified using other tools, additional steps may be required to further identify the RBridge that interfaces with the physical server.

It is important to have a set of tools that allow an operator to quickly and easily identify the physical MAC address associated with a given IP address, or IP addresses associated with a given physical MAC address. Additionally, it may be required to identify the RBridge that connects to the given IP address. In this section, we
present methods to identify MAC address to IP addresses or IP address to MAC address bindings.

Address binding tools presented here need to be exercised from either a router or an RBridge that has IP services enabled on a given VLAN.

There are two different address binding resolutions required

1. MAC address to IP addresses binding

2. IP address to MAC address binding.

We propose to use invARP [RFC 2390] to resolve MAC address to IP address(es) binding and ARP [RFC 826] to resolve IP address to MAC binding information. It is possible a given physical server to host multiple virtual machines (i.e. IP Addresses). Hence, it is expected to receive one or more responses, to an invARP request. However, invARP in its current form is incapable of identifying whether a single multi-homed host or multiple virtual hosts. At the time of RFC 2390 and original ARP standard RFC 892 were written, virtual machine concept did not exist. Hence, these protocols in its current form do not include virtual machine identifiers such as vNTAGs. This lapse of identification of virtual machines, make troubleshooting of large virtual machine networks, with dynamic server allocation, very difficult. Hence, we propose to extend, ARP [RFC 892] and invARP [RFC 2390], protocol to carry, virtual machine identification tags.

Upon discovery of MAC address or identification that a given MAC address is associated with a valid IP addresses, user may employ the locator utilities listed in section 9.7. to identify the corresponding RBridge and associated interface information. Alternatively, implementation may support ARP response snooping with extension explained in 9.5.1 to encode RBridge and location information into ARP or invARP responses.

9.6.1. Extension to ARP and invARP

RFC 2390 presents methods to discover protocol address associated with a given hardware address. In this section we propose methods to extend RFC 2390 and RFC 892 to encode additional virtual interface tag information and device information that may facilitate identifying physical machine locations.

It is important the extensions proposed in the standard are transparent to current implementations.
Figure 53, below, depicts the format of an ARP/invARP frame with the proposed extensions embedded.

ARP frame as defined in RFC 892 and RFC 2390 has a fixed structure and include only the length fields for addresses. Implementations index into these fix address fields and do not check the total length of the response frame as part of validation. Hence, we propose to include the extensions at the end after the target protocol address. Implementations that do not support the new extensions will safely ignore these values.

We expect additional identification information carried in ARP and invARP to be limited. Furthermore, these, identification information have compact and deterministic size. Hence, we propose not to use explicit, length identification field, instead derive the length of the value field implicitly, based on the class and class types defined below. ARP and invARP follow identical encoding structures.
9.6.1.1. Encoding ARP-invARP extensions

ARP Extension encoding structure and proposed extensions are presented in this section. We propose a compact structure for ARP encoding. In Figure 53 "Class" identifies the Object Class and the "Class Type" (c-Type) within the class identify specific data element within the object class. C-Type implicitly indicates the size of the object. The encoded object size MUST NOT exceed the implied size of the corresponding Class and c-Type.
Figure 53  Encoding of ARP Extensions

Class : (1 octet). Define to identify the Object Class.

C-Type : (1 octet). Define Object type within Object class.

Object : (Variable octet, depends on the Class and C-Types)

<table>
<thead>
<tr>
<th>Class</th>
<th>C-Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>vNTAG</td>
<td>vNTAG of the interface</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>rBridge</td>
<td>TRILL RBridge nickname</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>ifindex</td>
<td>ifindex of RBridge interface ARP</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Slot</td>
<td>Slot id of RBridge interface ARP</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Port</td>
<td>Port id of RBridge interface ARP</td>
</tr>
</tbody>
</table>

Figure 54  Table of Class, C-Type and usage

Figure 54, above, presents Class, c-Type and application definitions. vNTAG, rBridge, Slot and Port are each 2 octets in length. The length of ifindex is 4 octets. All of the above extensions are optional. vNTAG is inserted by the end station that is responding to the ARP request. All other fields are inserted by the TRILL RBridge that interface with the end-station and implement ARP response snooping. ARP response snooping is similar to Dynamic ARP inspections, implemented by many major vendors. Dynamic ARP inspection validates the Source IP address of ARP response against known IP addresses to prevent ARP cache poisoning by rogue stations. ARP response snooping, on the other hand, intercepts ARP response frames and inserts required fields as defined
in this standard. Implementations may extend the dynamic ARP inspection framework to implement ARP response snooping.

In the interim, most end stations and servers may not insert the proposed vNTAG information. Hence, optionally, ARP response snooping, process on TRILL RBridge, MAY insert vNTAG information on behalf of the end station or server.

9.7. End-Station Attachment Point Discovery

In traditional deployments, end stations and servers were relatively static in their locations. As a result localizing a fault was relatively easier.

The virtual machine concept is an increasing trend in Datacenter and large enterprises. Dynamic load balancing policies of Virtual infrastructure, based on various load balancing policies, move virtual machines between different physical servers. This dynamic motion of virtual machines causes difficulty in associating a given virtual server to a RBridge. As a result, localizing a fault is a difficult task and requires use of multiple applications. Some virtual machine deployments utilize a single MAC address to represent all the virtual servers in a single physical server. Hence, it is important, to identify both the physical attachment point and the virtual segment information, such as VLAN and Virtual Tags.

ARP/invARP extensions presented above facilitate discovery of the attachment information, however, some implementation may face scaling issues due to the large number of ARP requests. An alternative method is presented below.

The End-Station attachment Point Discovery methods presented here, allow discovering, RBridge, interface information, VLAN, virtual Tags, etc, associated with a given IP Address.

The End-Station attachment Point Discovery is a two step process. However implementations may present a single user interface that combines both the steps.

Step 1: Utilize ARP to discover the MAC address associated with the specified IP address. Identify the ingress RBridge nickname by analyzing the TRILL header and identify the VLAN information based on the inner VLAN.

Step 2: Utilize MAC discovery methods explained above to discover, interface and virtual Tag information associated with the MAC
address discovered in above Step 1. Implementation SHOULD narrow the scope of the MAC discovery to include only the RBridge and VLAN discovered in step 1.

9.8. DRB and AF Discovery

The TRILL Base Protocol standard [RFC 6325] specifies support for multi-access legacy network and shared segments between TRILL and legacy devices. Legacy networks ensure loop free forwarding via the IEEE 802.1D (Spanning Tree) protocol. RFC 6325 and RFC 6327 specify loop prevention methods in mixed environments where the TRILL network borders with a legacy multi-access network. RFC 6325 also provide methods for load splitting of native traffic in to the TRILL network. These are accomplished by having a single Designated RBridge (DRB) for a given LAN segment which designates an Appointed Forwarder (AF) for each VLAN on the segment to ingress and egress traffic originating and destined to and from the legacy network.

Based on network dynamics, configurations, and failures, DRB and/or AF designation may change from time to time. Hence, discovery of DRB and AF is very important to effectively troubleshoot network connectivity problems that involve TRILL and legacy networks connected via non P2P TRILL interfaces.

DRB-AF discovery message has three variations.

1. All DRB discovery
2. All AF discovery
3. VLAN,AF discovery

Above messages are identified with a unique TRILL OAM message code (section 8.).

DRB-AF discovery messages allow for identifying the following parameters:

- Nickname of the DRB
- STP Root Bridge identifier
- Up time of AF (if responder is the AF)
- Up time of DRB (if Responder is DRB)
- Enabled VLAN List
o Announcing VLAN List
o DRB State (If Responder is the DRB)
o AF State (If Responder is AF)
o Pseudo Node bypass (If the Responder is the DRB)
o Number of times the Designated VLAN has changed
o AF List (nickname,start VLAN,end VLAN)(If the Responder is DRB)

The above parameters are encoded in to the response message via ICMP multipart extensions (section 8.)

9.8.1. Theory of Operation

DRB-AF discovery message utilize same addressing and format as the MAC discovery message (Section 9.5.)

9.8.1.1. Originator RBridge

Follow the steps specified in section 9.5.1.1., with the following exceptions

Specify the message as one of the DRB-AF messages.

If the message is VLAN,AF discovery message, then include the interest VLAN list.

9.8.1.2. Receiving RBridge

Follow the processing steps specified in section 9.5.1.2. with the following exceptions:

If RBridge is in the scope list or All-RBridge scope is specified, then the RBridge processes the message as follows:

If the message is DRB discovery message then the receiving RBridge include the following information:

o Response code set to DRB discovery response (12)
o Nickname of the DRB
o Nickname of AF of the specified VLAN
o STP Root Bridge identifier

o DRB Life time

o Enabled VLAN List

o Announcing VLAN List

o DRB State

o Pseudo Node bypass

o Number of times Designated VLAN change

o AF List (nickname, start VLAN, end VLAN)

If the message is an AF discovery or VLAN, AF discovery message, then the receiving RBridge first validate whether the RBbridge is the AF for the specified VLAN list and include following information:

- Response code set AF discover response (14) or AF-VLAN discover response (16)
- Nickname of the DRB
- Nickname of AF of the specified VLAN or AF VLAN-List if VLAN is not specified.
- STP Root Bridge identifier
- AF Life time (i.e. How long has been AF)
- Enabled VLAN List
- Announcing VLAN List
- AF State
- Number of times Designated VLAN change

If RBridge is not the AF for specified VLAN then include ERROR code Not AF (4) (see Figure 27).
If RBridge is AF for only a subset of VLANs specified in the request then include WARNING "AF VLAN list Mismatch" (3) and include the VLAN list that the RBridge is functioning as AF. (Figure 28)

9.9. Diagnostic Payload Discovery for ECMP coverage

This document specifies that a 128 byte Diagnostic Payload to be embedded in the OAM frame. The Diagnostic Payload embedded in the OAM frame determines the ECMP path taken by the OAM frame. Hence, It is important to have methods that allow operators to discover diagnostic payload constructs that direct OAM frames through desired ECMP paths. RFC 4379 proposes a method to discovery payload combinations in MPLS networks. We propose to use a similar approach, with some modifications.

RBridge MUST derive diagnostic payload combination such that when applicable hashing methods are applied to the diagnostic payload, the OAM frames that contain the diagnostic payload follow the requested path. The diagnostic payload contain Destination and Source MAC addresses, VLAN Tag, Ethertype, Layer 3 and Layer 4 addressing information and packet data. TRILL RBridges operate as Layer 2 devices and learn source MAC addresses against ingress RBridge nickname. Use of any arbitrary MAC address as source MAC address may affect RBridge learning. Hence we suggest using either a well-known OAM MAC address or operator specified MAC address as the source MAC address of the generated diagnostic payload. TRILL, Layer 2 forwarding happens in the context of a VLAN. Specification of a random VLAN in the generated diagnostic payload may lead to different forwarding behavior of OAM frames than the actual data frames that operator desire to diagnose. Hence, operator is required to specify the desired VLAN in the payload generation request.

Operator generates Payload discovery command from RBridge RB(a), in the message operator MUST specify the seed Destination MAC address, desired VLAN and required ECMP coverage and final egress RBridge RB(x). Receiving RBridge (RBi) using the provided information in the request and using local hashing algorithm, generates series of proposed payloads. The generated payloads are returned to the requester. Requester may use the received proposal as a seed and request the next RBridge (RBj) downstream from RBi to generate diagnostic payloads that would cover the desired ECMP path downstream from RB(j). RB(a) may continue this process until specific set of payloads are derived such that it covers desired paths from ingress RBridge RB(i) to egress RBridge RB(x). These derived payload allows RB(a) to test end-end coverage from RB(a) to RB(x) over a specific path.
Encoding of proposed MAC address seed require further clarification and some illustration to ensure clearer understanding.

Seed MAC address is encoded in c-type 37 as 6 octet value. A given request can contain multiple seeds. Each of the seeds are indentified with a unique 12 bit identifier.

Each zero valued octet in a MAC address seed has a corresponding bit value vector (c-type 39). Non-zero octets of the MAC address seed are considered fixed valued and are not considered for payload proposal generation.

Bit value vector is 256 bits long. Each bit in the bit vector value represents a value for the corresponding octet of the MAC address seed. Values that are included in the proposal are represented by setting the corresponding bit vector values to 1.

As an Example let’s consider requester desire to use destination MAC address 0x00:0A:0B:00:00:00 to 0x00:0A:0B:0F:0F:0F to generate the payload proposals.

Requester encode the destination MAC address seed using c-type 37 (Seed Destination MAC address) as follows

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 1 1 0 1 0 1 0 0|0 0 0 0|0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 1 0 1 1|0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Corresponding desired range for each of the octets that contain 0 (zero) are encoded as follows, using c-type 39 (MAC octet bit vector).

Example encoding of MAC-0:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 1 1 0 1 0 1 0 0|0 0 0 0|0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Octet zero (MAC-0) of MAC address seed is represented in the above c-type 39. Value zero in MAC-0 field of the above c-type 39
indicates that other values may be considered for the proposal. However, in this example, for MAC-0 user requires maintaining value zero and does not desire for the responder to consider other options for MAC-0 field. Hence bit vector length is set to zero to indicate that.

Example encoding of MAC-3:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 1 0 0 1 0 1 0 0|0 0 1 1|0 0 0 0 0 0 0 0|0 0 0 1 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1 1 1 1 1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 0 0 0 0 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In the above example MAC-3 bit vector is represented using c-type 39. Bit vector offset has been set to zero to represent, propose value range start from 0. Bit vector length has been set to 16 to indicate 16 values from the bit vector offset to be considered for the proposal. In this example, the available range for the proposal is 0x0 to 0xF.

9.9.1. Theory of Operations

The ingress RBridge sends an Payload discovery OAM Command Message to the intermediate RBridge from which it desires to discover the diagnostic payloads for the specified ECMP choices. Also specified in the Command Message is the egress RBridge nickname, desired VLAN, EthType and ECMP choices.

As an example consider the topology in Figure 55. RB1 desires to identify diagnostic payloads required to cover all of the ECMP choices RB2 has towards egress RBridge RB7 for a specific VLANx.

RB1 generates an ECMP discovery OAM command message to RB2. In the ECMP discovery message, RB1 includes egress RBridge nickname (RB7), ECMP choices to be covered, interested VLAN (VLANx), Destination MAC address seed, and EthType.
9.9.1.1. Receiving RBridge

The receiving RBridge (RB2), first, MUST perform the required pre-processing and OAM message validation as specified in section 6.6.

Upon validation of the message, the receiving RBridge, using the ECMP selection algorithm of the local RBridge and the payload seed received from the requester, derives the required payload proposals for the requested ECMP choices such that OAM frames containing the proposed diagnostic payloads follow the requested ECMP path. If the received payload seed contain multiple seed values, the local RBridge is required to consider all of the seed values. Bit vector positions of the c-type 39 that do not generate the required ECMP choice or local RBridge did not consider for payload generation MUST be set to zero.

If the requested ECMP choice is not available at the Receiving RBridge, an ECMP selection error is generated. (e.g. Ingres RBridge requested to generate payload for path 10, and local RBridge has only 5 paths to the egress RBridge, then ECMP paths 6-10 are at error)

The resulting payload proposals are returned to the requester via Payload generation response OAM message. Payload generation response OAM message may be delivered either in-band or out of band to the requesting RBridge.
Following TLVs are required to specify the requested operations and results.

**TLVs in the Command Request Message**

- Payload Generation Request (c-type 40)
  - Egress RBridge nickname (c-type 40)
  - ECMP choices (c-type 40)
- Interested VLAN (c-type 4)
- Interested EthType (c-type 43)
- Seed Destination MAC Address (c-type 37)
- Seed Source MAC Address (c-type 38) (optional)
- MAC Octet bit-vector (c-type 39)
- Service Tag (c-type 23) (optional)

**TLVS in the Command Response Message**

- Payload generation response (c-type 41) (for each ECMP choice)
  - ECMP choice status (for each of the requested ECMP)
- Interested VLAN (c-type 4)
- Interested EthType (c-type 43)
- Seed Destination MAC Address (c-type 37)
- Seed Source MAC Address (c-type 38) (optional, included only if present in the request)
- MAC Octet bit-vector (c-type 39) (bit values appropriately set)
- Service Tag (c-type 23) (optional)

Please see section 8.2. for encoding format of the applicable TLVs.

The request originator may utilize the above payload proposals received from an intermediate RBridge to iteratively discover payload proposals along the path from ingress RBridge to the desired RBridge. At each of the iteration requester may utilize received proposals as seeds to the next hop downstream RBridge.

**9.10. Notification Messages**

Notification messages are generated either due to regular TRILL data frames or TRILL OAM frames. Implementation MUST not generate notification messages on notification messages.
There are 3 types of Notification messages:

- Time Expiry
- Destination Unreachable
- Parameter Problem

Within these Notification messages, error, warning and information ICMP extensions may be included to identify the details of the notification message. Section 4.3. above covers details of encoding Notification messages, section 8.2. covers ICMP extensions.

Time expiry messages are generated when TRILL hope-count field reach to zero. If applicable, it may contain additional error, warning or information extensions.

Destination unreachable notification may be generated for following scenarios; additional scenarios may be added later.

- Egress RBridge nickname unknown
- Inner VLAN does not exist or suspended
- Not the AF for inner VLAN

Parameter Problem notification may be generated for following scenarios; additional scenarios may be added later.

- Invalid RBridge nickname (RBridge nickname is one of the reserved 0xFFC0 - 0xFFFF)
- MTU mismatch
- Invalid VLAN (Reserved VLANs)
- Interface state is not forwarding

10. Monitoring and Reporting

Proactive identification of data plane failures are important part of maintaining Service Level Agreements (SLA). In traditional Layer 2, networks, there is only a single active path to monitor and both multicast and unicast traffic follow identical paths. With TRILL, there are multiple active paths and unicast and multicast traffic take potentially different paths, depending on the flow parameters.

TRILL deployment in a typical data center may have 10’s of 1000 of links and 100’s of RBridges. In such an environment, there may be large number of active paths between two end points. As an example, assume a topology with 4 RBridges connected serially with 32 ECMP links at each hop. In the stated example topology, there are
32x32x32=32768 possible paths. Monitoring all of the possible path combinations is not scalable. However, skipping some combination of paths leads to reduce coverage and hence reduced effectiveness of monitoring data. Even if one was brave enough to monitor all of the links, analyzing and diagnosing a problem is quite cumbersome due to the large amount of data. In other words, there must be methods to scale the problem and present information in a more concise manner that is still effective.

In this document we propose to use the "region" concept to partition the network into logical sections. Regions are monitored independently. Detailed sets of monitoring data are distributed throughout the region. A summary set of monitoring data is distributed throughout the network. Network operators can obtain a network health snapshot of the entire network from any RBridge in the network. Detailed health report of a given region can be obtained from any RBridge in the region.

An RBridge associate itself with a region through its interfaces. A given interface can belong to one and only one region. An RBridge can have multiple interfaces belonging to different regions. Each RBridge is responsible for collecting monitoring data, organizing the data into regions and advertising the data to its peers. Please see section 10.2, Advertising Policy for details.

In theory a network topology can be any arbitrary graph. In practices, however, it is some set of sub-graphs repeating to construct the overall topology. Each sub-graphs or set of sub-graphs can be considered a region for monitoring purpose. The manner in which regions are partitioned is an administrative choice such that;

1. Maximize the fault coverage.

2. Optimize network health data summarization.

As an example consider a typical datacenter topology depicted in Figure 10. Typical datacenter may have multiple Points of Demarcation (POD)s connected with an aggregation layer. A POD can be considered as a region and may be individually monitored.
Figure 56   Example of "regions"

10.1. Data categories

There are 3 categories of monitoring data. They are, Summary Category, Detail Category and Vendor Specific Category. The Summary and Detail categories are mandatory. That is, every RBridge that is compliant to this standard and support Monitoring, MUST support all the elements defined under the Summary and Detail categories. The Vendor specific Category is optional. Vendor specific data elements are only available within the region. An RBridge that does not understand the Vendor specific data elements forward them to neighboring RBridges per Advertising Policies define in section 10.2. Individual data elements and structure of encoding Summary, Detail and Vendor specific categories are presented in sections 10.3. - 10.5.

10.2. Advertising Policy

Each RBridge is responsible for advertising monitoring data to the OAM capable neighbors.
Different interfaces on an RBridge can belong to different regions. However, a given interface can belong to one and only one region. As a result, a given RBridge may receive data from multiple regions. Each RBridge is responsible for advertising proper data categories over a given interface to the neighbor.

Rule 1: No monitoring data are distributed:
- On legacy interfaces
- To neighbors not OAM capable
- When ISIS state is not 2-way
- When monitoring data advertisement is disabled

Rule 2: Distribution of Summary category data:
- Distribute on all OAM capable interfaces
- Do not distribute summary data element of a region back to the originating region. (i.e. do not distribute on to interfaces that have the same region name as the data element)
- Summary data for local region is derived from Detail data. (local summary data is never advertised into the local region per the above rule. However, it is advertised out to other regions the RBridge has interfaces in to)

Rule 3: Distribution of Detail category
- Distributed on OAM capable interfaces
- Region of the data element and region of the interface must match for propagating a data element over an interface (i.e. Do not advertise to other regions)
- Do not advertise data element back in to the originator RBridge.

Each RBridge distributes data at periodic intervals. Each RBridge collects data it has received, analyzes them, and redistributes according to the rules specified above. The distribution interval should be appropriately adjusted to not overload ISIS routing operations.
Monitoring application is responsible for maintaining the Application specific LSP. We propose to use Generic Application Encoding methods explained in [GenAPP] for distributing Monitoring data. TRILL operates in ISIS Level-1 layer, hence S,D flags defined in [GenAPP] MUST be set to zero.

We propose to obtain specific Application ID [GenAPP][RFC5226] from IANA for the purpose of registering TRILL Monitoring data distribution.

Within the Application ID context, a series of sub-TLV are defined to carry specified information.

10.2.1. Multi Instance ISIS and Flooding Scope

As presented above, Summary data has a flooding scope of the entire ISIS domain and Detail and Vendor data have a flooding scope of the applicable monitoring region.

[ISISMI] provides a frame work to define multiple instances of ISIS and multiple instances of ISIS topologies within a given ISIS instance. These topologies may have different flooding scopes. The flooding scope of a topology limits the extent of the distribution of an LSP associated with that topology. Topologies defined within the ISIS TRILL-OAM instance are independent of the TRILL data plane multi-topology definitions within the TRILL ISIS protocol instance.

It is recommended to have a separate ISIS instance for the purpose of TRILL-OAM. Within the TRILL-OAM ISIS instance, the following topologies MUST be defined with the specified flooding scope.

The Global Topology is created within the TRILL-OAM ISIS instance to include all of the RBridges in the OAM domain. Summary category GenAPP data LSPs are flooded within the scope of the Global Topology.

Regional Topologies are created within the TRILL-OAM ISIS instance per each region for regions a given RBridge is associated with. A Regional Topology includes RBridges and interfaces within the applicable region. LSPs carrying Detail and Vendor category data are flooded within the applicable Regional Topology.

10.3. Summary Category

Then following individual data elements are defined within the summary category.
o Name of the region

o Total number of R Bridges in the regions

o Total number of TRILL enabled ports in the region

o Percentage of TRILL enabled ports down

o Percentage of TRILL enabled ports oversubscribed

o Maximum number of paths in the largest ECMP in the region

Then following structure encodes each of the data elements within the summary category.

```
+----------+
| subTlv   | 2 octets
+----------+
| Region-ID | 4 octets
+----------+
| L |                  |
+---+                  
    |                     |
    |                     |
    +----------+
| #Rbridges | 2 octets
+----------+
| #Ports    | 2 octets
+----------+
| #UpPorts  | 2 octets
+----------+
| #OsubPort | 2 octets
+----------+
| #ErrPorts | 2 octets
+----------+
| #ECMP     | 2 octets
+----------+
| #DwnPorts | 2 octets
```

Figure 57   Encoding Summary Category Data

subType : (2 octets) is always 1 for summary category
Region-ID : (4 octets) is unsigned 32 bit integer identifier of the region

L       : ( 1 octet), length of the subsequent field

Region Name : ‘\0’ terminated ASCII string of region name of variable size to maximum of 255 octets.

#Rbridge: (2 octets), number of RBridges in the region

#Ports: (2 octets) Total number of TRILL enabled ports available on this RBridge

#Up Ports: (2 octets) Total number of TRILL enabled ports that are operationally up.

#OSPorts : (2 octets) Total number of TRILL enabled ports that are oversubscribed.

#ErrPorts : (2 octets) Total number of TRILL enabled ports that are indicating errors.

#DwnPorts : (2 octets) Total number of TRILL enabled ports that are operationally down.

#ECMP : (2 octets) Maximum number of ECMP as seen by this region ISIS routing table.

10.4. Detail Category

Following data elements MUST be present within the detail category.

- Name of the region
- Name of the RBridge
- RBridge up time
- Total number of neighbors
- Total number of TRILL enabled ports in the RBridge
- Total number of TRILL enabled ports Up
- Total number of TRILL enabled ports oversubscribed
o Total number of TRILL enabled ports observing errors

o Maximum number of links in the largest ECMP of the switch

o Port data: Name of each TRILL enabled Port and Port state (Up, oversubscribed, error) and interface index.

o Adjacency Matrix

  o List of (neighbor RBridge nickname and interface index of ports connecting to the neighbor RBridge).

  o NOTE: Interface index in the Adjacency matrix is used as key in to port data to obtain Port name and state.
Figure 58  Encoding Detail Category Data

subType : (2 octets) always 2 for Detail category

RBridge: (2 octets) TRILL RBridge nickname [RFCtrill]
Region-ID : (4 octets) unsigned 32 bit integer identifier of the region

L     : ( 1 octet), length of the subsequent field

Region Name : ‘\0’ terminated ASCII string of region name of variable size to maximum of 255 octets.

Up Time: (8 octets), number of seconds RBridge has been operational. If an RBridge reaches maximum count, it MUST NOT rollover.

#Ports: (2 octets) Total number of TRILL enabled ports available on this RBridge

#Up Ports: (2 octets) Total number of TRILL enabled ports that are operationally up.

#OSPorts : (2 octets) Total number of TRILL enabled ports that are oversubscribed.

#ErrPorts : (2 octets) Total number of TRILL enabled ports that are indicating errors.

#DwnPorts : (2 octets) Total number of TRILL enabled ports that are operationally down.

#ECMP : (2 octets) Maximum number of ECMP as seen by this RBridge ISIS routing table.

subtype-2: (2 octets): Set to 3. Following this sub type is the variable length Port Data. See below for details

sutype-3: (2 octets): Set to 4. Following this sub type is the variable length Adjacency Matrix. See below for details
<table>
<thead>
<tr>
<th>subType</th>
<th>2 octets</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBridge</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>F</td>
<td>1 octet</td>
</tr>
<tr>
<td>subtype-p</td>
<td>2 octets</td>
</tr>
<tr>
<td>ifindex</td>
<td>4 octets</td>
</tr>
<tr>
<td>Slot</td>
<td>Port</td>
</tr>
<tr>
<td>Speed</td>
<td>State</td>
</tr>
</tbody>
</table>

Figure 59 Encoding Port data

subType : (2 octets) Set to 3 for Port Data

RBridge: (2 octets) TRILL RBridge nickname [RFCtrill]

Region-ID : (4 octets) unsigned 32 bit integer identifier of the region

L  : ( 1 octet), length of the subsequent field in octets.

Region Name : ‘\0’ terminated ASCII string of region name of variable size to maximum of 255 octets.

F : (1 octet) Flag. When set, indicates this is the last Port data set from this node. It is possible Port data encoding to exceed MTU size due to large number of interfaces. The F flag allows to for advertising the information in multiple LSP packets.

subtype-p: (2 octets) set to 5 to indicate that this is a single Port entry within subtype 3. SubType 5 MUST always be embedded with subtype 3. Within subtype 3 there can be multiple subtype 5, one for each port entry.

Ifindex : (4 octets) 32 bit unsigned integer, used as key to port data advertised.

Slot   (2 octets)  : Slot number

Port   (2 octets)  : Port number
Speed    (2 octets) : Speed in 100Mbps. Zero (0) indicates port speeds less than 100Mbps.

State    (2 octets) : Represent the state of the port.

0: Down - no errors
1: Disable
2: Forwarding-no errors
3: Down - errors
4: Forwarding - errors
5: Forwarding - oversubscribed
6: Link Monitoring disable
All other values reserved.

+----------+
| subType  |            2 octets
+----------+
| RBridge  |
+----------+
| F        |            1 octets
+----------+
| subtype-a|            2 octets
+----------+
| nrBridge |            2 octets
+----------+
| #ports   |            2 octets
+----------+
| ifindex  |            4 octets
+----------+

Figure 60   Encoding Adjacency Matrix

subType : (2 octets) set to 4 for Adjacency Matrix

RBridge: (2 octets) TRILL RBridge nickname [RFCtrill]

Region-ID : (4 octets) unsigned 32 bit integer identifier of the region

L   : ( 1 octet), length of the region name in octets
Region Name: ‘\0’ terminated ASCII string of region name of variable size to a maximum of 255 octets.

F: (1 octet) Flag. When set, indicates this is the last Port data set from this node. It is possible Port data encoding to exceed MTU size due to large number of interfaces. The F flag allows to for advertising the information in multiple LSP packets.

subtype-a: (2 octets) set to 6 to indicates a single adjacency entry within subtype 4. SubType 6 MUST always be embedded with subtype 4. Within subtype 4, there can be multiple subtype 6, one for each adjacency.

nrBRIDGE: (2 octets), nickname of the next hop RBridge

#ports: (2 octets), total number of parallel links from RBridge to nrBRIDGE

Ifindex: (4 octets) 32 bit unsigned integer, used as key to port data advertised.

10.5. Vendor Specific Category

Vendors may specify additional data elements to be distributed as part of the monitoring data suite. All vendor specific data elements MUST contain the regions name and follow the structure defined below.
### Figure 61  Encoding Vendor specific category Data

<table>
<thead>
<tr>
<th>subType</th>
<th>2 octets</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBridge</td>
<td>2 octets</td>
</tr>
<tr>
<td>Region-ID</td>
<td>4 octets</td>
</tr>
<tr>
<td>L</td>
<td></td>
</tr>
<tr>
<td>Region Name</td>
<td></td>
</tr>
<tr>
<td>Vendor OUI</td>
<td>4 octets</td>
</tr>
<tr>
<td>Vendor specific Information</td>
<td></td>
</tr>
</tbody>
</table>

subType : (2 octets) set to 250 for Vendor specific category

RBridge: (2 octets) TRILL RBridge nickname [RFCtrill]

Region-ID : (4 octets) unsigned 32 bit integer identifier of the region

L : (1 octet), length of the region name in octets

Region Name : ‘\0’ terminated ASCII string of region name of variable size to maximum of 255 octets.

Vendor OUI : 3 octets of IEEE vendor OUI. Right justified. Most significant octet in network byte order is set to zero and ignored on receipt.

Vendor specific information : variable size and vendor dependent.

### 11. Traffic Triggered Monitoring (TTM)

Identification and verification of faults as well as fault monitoring methods using simplified payload structures were presented in previous sections of this document. In practice some faults may be due to more complex relationship between several
flows. The Traffic Triggered Monitoring methods presented in this
section proposes methods to monitor and analyze traffic passing
through different Test Points (TP) in the network. Additionally,
some of the methods presented earlier require having one or more
fields of the payload to be fixed to some known pattern. Use of
known patterns in payloads, while adequate in many occasions, may
not be adequate in other occasions. TTM allows operators to monitor
and/or diagnose a network using actual live traffic, with minimum or
no impact on actual data flow. The TTM Framework has the following
components.

TTM Profile: Is bound to a TTM Test Point (interface). Specify the
structure of the data stream (i.e. MAC, IP address, VLAN etc) that
need to be monitored and associated actions, frequency and duration.

TTM Initiator: An RBridge or external station that initiates a TTM
profile.

TTM Receptor: An RBridge that installs and monitors TTM Profiles on
a TP on behalf of a TTM Initiator.

TTM Test Point (TP): An interface on a specified RBridge.

TTM Messages: TTM Messages provides a messaging framework for TTM
related inter RBridge communications. The TTM messaging framework is
an extension to the OAM command messages.

TTM ingress End Point: The TTM ingress End Point is the ingress
RBridge of the specified flow.

TTM egress End Point: The TTM egress End Point is the egress RBridge
of the specified flow.
11.1. TTM Policy

The TTM policy is a high level container that defines rules and actions. The TTM policy contains several sections.

- TTM pattern
- TTM mask
- TTM Class
- TTM frequency
- TTM count
- TTM actions
- TTM Test Point
- TTM Ingress Point
- TTM Egress Point

TTM pattern: The TTM pattern can be either 128byte opaque data or set of fixed fields. The 128byte opaque data section allows users to define a required pattern. The TTM fixed fields are Dest MAC, Src MAC, VLAN, EthType, Src IP, Dest IP, TTL, Protocol Type, or Src/Dest UDP ports.

TTM mask: The TTM mask allows users to further refine the pattern matching criteria.
TTM Class: The TTM Class defines whether the TTM policy is Forward Flow Monitoring (FFM) or Reverse Flow Monitoring (RFM). Please see below for details.

TTM frequency: TTM frequency defines the frequency of actions specified.

TTM Count: TTM Count defines number of times the given TTM actions such as Capturing, Logging, Sampling and Injecting must be applied. Count is a 32bit unsigned integer. 1 indicates single instance. 0xFFFF indicates continued application until the TTM is removed by user actions.

TTM actions: TTM actions are

- count RX frames
- count TX frames
- count RX bytes
- count TX bytes
- count errors,
- log,
- Capture etc.

NOTE: TTM action counters are 64bits wide. Counter values may be distributed using the distribution framework, specified in section 10. Distribution of counter values allows user to monitor statistics from any remote RBridge.

Logging indicates logging a copy of the received frame in to a locally defined space.

Capture indicates forwarding a copy of the frame matching the TTM policy to a remote destination.

Implementation of logging and capture are outside the scope of the document.

TTM Test Point: TTM Test Point is an interface on an RBridge where the specified TTM profile is applied. User may either specify one or more interfaces or specify automatic. The automatic scope indicates the Receptor RBridge will derive the Test Points using ingress and egress End Point specifications.

TTM ingress End Point: TTM ingress End Point is the nickname of the ingress RBridge.
TTM egress End Point: TTM egress End Point is the nickname of the egress RBridge.

11.2. TTM Commands

TTM commands:

- TTM Set
- TTM Get
- TTM Remove
- TTM Response
- TTM Indications

TTM Set message is OAM Message type 17. This message is originated by the Initiator to install a TTM profile.

TTM Get message is OAM Message type 18. This message is originated by the Initiator to Get a TTM profile or sub component of a profile such as a counter.

TTM Remove message is OAM Message type 19. This message is originated by the Initiator to Remove a TTM profile.

TTM Response message is OAM Message type 20. This message is originated by the Receptor in response to one of the Set, Get or Remove messages. The Response message contains a message sub-code to indicate whether it is a response to a Set, Get or Remove message. It also contains the status code of the original request.

TTM Indications are generated by the receptors in response to asynchronous events such as packet capture.

TTM policies are encoded in to the OAM command messages using structures defined in section 8.2.

Forward Flow Monitoring (FFM)

The exact path taken by a given frame depends on the pattern of the payload. Forward Flow monitoring allows users to specify TTM profiles that match a specified policy in the direction of the normal traffic flow. i.e. Traffic ingress from the TTM ingress End Point and egress from the TTM egress End Point.
11.3. Reverse Flow Monitoring (RFM)

Traffic is bi-directional in nature. Any effective OAM solution should have methods to detect and monitor traffic flows in both forward and reverse directions. RFM allows users to:

1. Monitor frames traversing in the reverse direction. That is frames traversing from TTM egress End Point to TTM ingress End Point.

2. Inject a given data frame from a specified RBridge (RBe) to (RBi). The TTM policy contains additional user data field that specify the frame that is to be injected from RBe to RBi.

12. Security Considerations

Security considerations are under investigation.

13. IANA Considerations

13.1. IANA considerations

Following IANA considerations are required

13.1.1. ICMP Extensions

Request IANA to assign new Class-Num for TRILL OAM ICMP extensions.

Request to form a sub-registry under ICMP extensions to include c-types defined in this document and allocate future requests. Currently c-types 1-20 are defined in section 8.2.

13.1.2. TRILL-OAM UDP port

Request IANA to assign a well-known UDP port for the purpose of TRILL-OAM. Details of usage of well-known UDP port are presented in section 4.3.1.

13.1.3. ARP Extensions

Request IANA to form a new registry to allocate ARP extensions defined in section 9.6.1. Class-Num allocated within ARP extensions are allocated by IANA on first come first serve basis. C-type within a given Class-Num are defined by owners of the Class-Num and sub-registry MUST be established within ARP extensions.
13.1.4. Well known Multicast MAC

Request IETF authority to allocate one of the TRILL allocated Multicast MAC address (01-80-C2-00-00-43 to 01-80-C2-00-00-4F) for the purpose.

13.2. IEEE Registration Authority Consideration

Well known unicast MAC address for the purpose of identifying OAM frames.

Well known unicast MAC address for the purpose of identifying certain OAM frames.

EthType <TBD> for the purpose of identifying OAM frames.

14. References

14.1. Normative References


14.2. Informative References


15. Acknowledgments

Authors wish to thank people who volunteered to review this document and provided comments. Les Ginsberg provided guidance, comments and support in defining usage of GenApp and ISIS-MI. Carlos Pignataro and Naiming Shen provided valuable comments related to ICMP extensions.

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

A.1. Sample Reports

In this section we present sample reports of summary data and sample output of detail data.

A.2. Summary Report

<table>
<thead>
<tr>
<th>Region</th>
<th>Number of switches</th>
<th>Max ECMP</th>
<th>Total# Of Ports</th>
<th>% of Up Ports</th>
<th>% of Ports Oversubscribed</th>
<th>Err Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxx</td>
<td>40</td>
<td>16</td>
<td>400</td>
<td>100</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>yyy</td>
<td>8</td>
<td>2</td>
<td>25</td>
<td>75</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>
### A.3. Detail Report

Region Name : <xx>

- Total Number of Switches in the region : 10
- Total Number of Core Ports in the region : 16
- Number of Operationally up Core Ports : 14
- Number of Oversubscribed Core Ports : 2
- Number of Error Core Ports : 0

- Maximum Switch Up Time : 15days:8Hr:10M:0S
- Minimum Switch Up Time : 0days:0Hr:1M:0S

Switch Adjacency Matrix:

- (*) oversubscribed Links
- (x) down Links
- (?) error Links

<table>
<thead>
<tr>
<th>Switch</th>
<th>Next Hop switch</th>
<th>Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>S2</td>
<td>eth81,eth8/2(*) , eth81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eth 10/2(x)</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>eth5/1 (?)</td>
</tr>
<tr>
<td></td>
<td>S4</td>
<td>eth5/2, eth7/1</td>
</tr>
<tr>
<td>S2</td>
<td>S1</td>
<td>eth4/1, eth4/2, eth3/1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eth3/2(x)</td>
</tr>
</tbody>
</table>
A.4. C-Type usage in messages

The Table below lists various OAM messages and applicable mandatory and optional c-types.

<table>
<thead>
<tr>
<th>Message</th>
<th>Mandatory Parameters</th>
<th>Optional Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loopback Request</td>
<td>Version (1)</td>
<td>VLAN (4) Service Tag (23)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Out-of-band request Flag (1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reverse Path (26) Control Plane</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Verification (24) Originator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IP address (2)</td>
</tr>
<tr>
<td>Loopback Response</td>
<td>Version (1)</td>
<td>Reverse Path Response (27)</td>
</tr>
<tr>
<td></td>
<td>Cross Connect Error</td>
<td>Control Plane Response (25)</td>
</tr>
<tr>
<td></td>
<td>Flag (1)</td>
<td>Final Flag (1)</td>
</tr>
<tr>
<td>Path Trace Request</td>
<td>Version (1)</td>
<td>VLAN (4) Service Tag (23)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Out-of-band request flag (1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reverse Path (26) Control Plane</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Verification (24) Originator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IP address (2)</td>
</tr>
<tr>
<td>Path Trace Response</td>
<td>Version (1)</td>
<td>Reverse Path Response (27)</td>
</tr>
<tr>
<td></td>
<td>Cross Connect Error</td>
<td>Control Plane Response (25)</td>
</tr>
<tr>
<td></td>
<td>Flag (1)</td>
<td>Final Flag (1)</td>
</tr>
<tr>
<td></td>
<td>Upstream Identification (2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Downstream Identification (5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Path of this Payload (6)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 63  Optional and Mandatory c-types
<table>
<thead>
<tr>
<th>Message</th>
<th>Mandatory Parameters</th>
<th>Optional Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast Tree Verification Request</td>
<td>Version (1)</td>
<td>VLAN (4)</td>
</tr>
<tr>
<td></td>
<td>Service Tag (23)</td>
<td>Control Plane</td>
</tr>
<tr>
<td></td>
<td>In Scope (14)</td>
<td>Verification (24)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Originator (24)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IP address (2)</td>
</tr>
<tr>
<td>Multicast Tree Verification Response</td>
<td>Version (1)</td>
<td>Control Plane</td>
</tr>
<tr>
<td></td>
<td>Cross Connect Error Flag (1)</td>
<td>Response (25)</td>
</tr>
<tr>
<td></td>
<td>Final Flag (1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Upstream Identification (2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multicast Tree</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Downstream List (15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RBridge nickname(35)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 64  Optional and Mandatory c-types

Authors’ Addresses

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

Phone: 408-853-2291  
Email: tsenevir@cisco.com
OAM tool for R Bridges: Multi-destination Ping

draft-yizhou-trill-multi-destination-ping-01.txt

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on April 31, 2009.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Abstract

Unicast and multi-destination data frame may follow the different path in TRILL network. We need the ping and traceroute like applications for the connectivity testing and fault isolation on the multi-destination path in addition to the unicast path. This document specifies the format and handling of the new TRILL OAM protocol messages and TLVs which can be used for the multi-destination OAM.

Table of Contents

1. Introduction ................................................ 3
2. Conventions used in this document............................ 3
3. Motivations ................................................. 3
4. RBridge Channel Message Format................................ 4
5. OAM Protocol Frame Format for Echo in the Long Format ....... 4
6. TLV Encodings ............................................... 6
   6.1. Target RBridge ......................................... 6
   6.2. Jitter ................................................. 7
7. Processing Echo Messages for Multi-destination Path........ 8
   7.1. Sending an echo request................................. 8
   7.2. Receiving an echo request................................ 9
       7.2.1. If H Flag is Not Set..................................... 9
       7.2.2. If H Flag is Set.................................... 10
   7.3. Sending an echo reply.................................... 11
   7.4. Receiving an echo reply.................................. 12
8. Security Considerations..................................... 12
9. IANA Considerations ........................................ 12
10. References ................................................ 12
   10.1. Normative References.................................. 12
   10.2. Informative References................................ 13
11. Acknowledgments ........................................... 13
1. Introduction

When RBridges are deployed in a real network, a number of applications are necessary for error detection/reporting and diagnostic purpose. TRILL RBridge channel [RBridgeChannel] was designed for carrying the OAM relevant messages. [RBridgeOAM] has defined the ping and traceroute applications for unicast path and also the error reporting mechanisms.

Multi-destination data path in TRILL network has different characteristics from the unicast path. One or more distribution trees are formed for multi-destination traffic. RBridges advertise their interests in receiving the traffic of the specific VLANS. The distribution tree may or may not be pruned based on VLAN ID. Troubleshooting on the multi-destination path is a desirable feature of TRILL OAM. This document specifies the messages and mechanisms used by multi-destination OAM.

2. Conventions used in this document

The same terminology and acronyms are used in this document as in [RF6325].

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].

3. Motivations

In an RBridge campus, unicast and multi-destination traffic may follow different paths between the same ingress and egress RBridges. [RBridgeOAM] specifies some OAM along unicast path. For diagnostic purposes it is also desirable to check the connectivity between two or more RBridges along a particular distribution tree.

There are various things we want to test for multi-destination path.

- Along a distribution tree, who are the leaf nodes of an inner VLAN? The leaf nodes here refer to the RBridges announcing the given inner VLAN as their interested VLAN in INT-VLAN sub-TLV. It is useful when we want to check if the configuration/provisioning are consistent with the design.

- Along a distribution tree, check the connectivity from the ingress RBridge to one or more leaf nodes of an inner vlan. It can be used as the first step in diagnosis when we suspect multi-destination data path to certain RBridge fails. Transit nodes do not decapsulate the
multi-destination data frame; therefore we do not think it is much of interest to check the connectivity to any non-leaf RBriges.

- Along a distribution tree, trace the multi-destination data path hop-by-hop to a target RBridge. It is useful when we want to find out where exactly is the failed hop.

This document specifies new messages and TLVs used by multi-destination OAM applications like multi-destination ping and traceroute. Processing of these messages is also discussed in the draft.

4. RBridge Channel Message Format

The RBridge Channel Header fields is as follows,

- CHV (Channel Header Version): zero.
- Channel Protocol: 0x006 (Echo in the Long Format) (TBD)
- Flags: The SL and NA bits SHOULD be zero, the MH bit SHOULD be one
- ERR: zero.

5. OAM Protocol Frame Format for Echo in the Long Format

The frame format is shown as follows. In the rest of this document, echo request and echo reply are brief ways to refer to the Echo Request in Long Format and Echo Reply in Long Format messages.
Figure 1 Echo Request with Long Format

- Sender’s Instance: An instance ID used by sender to associate the echo operation with different application instances, e.g. different Telnet sessions. Echo reply should return the value unchanged.
- SPID:
  1 - Echo Request in the Long Format
  2 - Echo Reply in the Long Format
- Sequence Number: An arbitrary 28-bit unsigned integer used to aid in matching reply messages to echo requests. It MAY be zero.
- Reply Mode: Default is 2. It can take one of the following values.
  1 - Do not reply. It can be used for one-way connectivity check. The receiving RBridge may perform monitoring and statistics collection on delay and/or jitter using one-way echo operation.
2 - Reply with Echo Reply in the Long Format and send back unicast in TRILL OAM channel. This value would be used by echo request in most cases.

- Flags: A bit vector with the following format. Currently only the H (Respond Only When Hop Count is Zero) flag is defined. In practice, we set H flag to be 0 for ping type applications and 1 for traceroute type applications of multi-destination OAM. With H flag set, it will help to prevent the duplicate echo replies from the same RBridge triggered by echo request with different hop count value in the same traceroute operation. H flag is only significant in echo request and MUST NOT be set in echo reply. The detailed processing based on the value of H flag is explained in section 7.2.

```
| 0| 1| 2| 3| 4| 5| 6| 7|
+--+--+--+--+--+--+--+--+
|     MBZ            | H|
+--+--+--+--+--+--+--+--+
```

- TimeStamp Sent: time-of-day (3 octets for seconds and 3 octets for microseconds) in NTP format that the echo request was sent according to the sender’s clock.

- TimeStamp Received: time-of-day (3 octets for seconds and 3 octets for microseconds) in NTP format that the corresponding echo request was received according to the receiver’s clock. This value is significant only in echo reply and MUST be set to all zeros in echo request and ignored on receipt of an echo request.

- TLVs: A set of type, length, value encoded fields as specified in next section.

6. TLV Encodings

6.1. Target RBridge

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|     Type = 0x05       |  Length = 2 + 2*n     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|           Number of Target RBridges           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
.              Target RBridge Nickname 1        .
.                     ...                       .
.              Target RBridge Nickname n        .
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```
o **Number of Target RBridges:** The number of nicknames specified in the following fields, the maximum number is 127.

o **Target RBridge Nickname:** The nickname of a Target RBridge.

This TLV MAY appear in an echo request. It SHOULD be copied back in the corresponding echo reply messages.

Target RBridge TLV is used by multi-destination OAM. For ping along the multi-destination path, the Target RBridge TLV with multiple nicknames MAY be included in echo request. It implies RBridges with any of the nicknames in the TLV should reply. While for traceroute like application, only a single nickname can be included in this TLV. If there was more than one nickname in the TLV, only the first nickname MUST be used as target nickname for tracing purpose.

When Target RBridge TLV is not included in an echo request, it implies the unspecified target. If an echo request with unspecified target was sent by ping like applications, then all leaf nodes in distribution tree pruned by the given inner VLAN SHOULD send back echo reply. If echo request with unspecified target was sent by traceroute like application, RBridges receiving the incoming frame with hop count value 1 would process the echo request and send back 'Hop Count is Zero' error notification.

### 6.2. Jitter

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>+-----------------------------------------------+-----------------------------------------------+-----------------------------------------------+-----------------------------------------------+-----------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type = 0x07</td>
<td>Length = 0x02</td>
<td>Jitter time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Jitter time:** Set to the upper bound of the jitter period in milliseconds. A responding node SHOULD wait a random amount of time between zero milliseconds and the value specified.

This TLV MAY appear in an Echo Request in the Long format. It SHOULD NOT be present in echo reply messages.
7. Processing Echo Messages for Multi-destination Path

7.1. Sending an echo request

The inner frame header and TRILL header fields are as follows:

- Inner.MacSA: MAC address of RBridge originating the echo request
- Inner.MacDA: Defaults to All-Egress-RBridges. It can be set to L2 multicast address derived from IP multicast group.
- Inner.VLAN ID: Defaults to 1. It can be any enabled VLAN ID on the ingress RBridge.
- Ingress RBridge Nickname: the nickname of RBridge originating the echo request
- Egress RBridge Nickname: the nickname of a distribution tree root
- M bit: 1
- Hop Count: defaults to maximum value 0x3F.

- For ping like applications, it can be any value which is believed to be no less than the number of hops from ingress RBridge to the most distant target RBridge in the tree.

- For traceroute like applications, hop count value starts from 1 and is increased by one for each sending of echo request.

H(Respond Only When Hop Count is Zero) flag in echo request is set to 1 for traceroute like applications and 0 for ping like applications.

The originating RBridge chooses the values of Sender’s Instance and Sequence Number for the echo request. Sequence number should be increased by 1 for each new subsequent echo request of the same Sender’s Instance. The Timestamp Sent is set to the time-of-day in NTP format [NTP] according to the sender’s clock. The Timestamp Received is set to zero.

The originating RBridge MAY use Target RBridge TLV to specify the target. For ping like applications, multiple nicknames MAY be present in one such TLV if sender wants to ping multiple targets at one time. For traceroute like applications, the TLV should at most contain one nickname as the tracing target. If there is more than one nickname, only the first one takes effect.
Echo request without Target RBridge TLV means the originating RBridge potentially wants to target every RBridge in the distribution tree. We also call it echo request with the unspecified target. For ping like applications, echo request with the unspecified target implies the sender wants to know who are the leaf nodes of the inner VLAN in the distribution tree. For traceroute like applications, it implies the sender wants to know the whole distribution tree structure hop-by-hop.

The Originating RBridge MAY include the Jitter TLV (see section 6.2) in the echo request in order to randomize the delay of the replying echo message from multiple RBridges.

7.2. Receiving an echo request

RBridge receiving an echo request with M bit set with EtherType of RBridge channel [RBridgeChannel] SHOULD replicate it to the control plane for processing and also forward it as normal multi-destination data frame. When a RBridge receives an incoming frame with hop count is 1 in TRILL header, it will not forward the frame further. If reply mode is 1, no echo reply is generated. For the sub sections below, we assume the reply mode is set to 2.

7.2.1. If H (Respond Only When Hop Count is Zero) Flag Is Not Set

- If Target RBridge TLV is not present in the echo request:

  All leaf nodes of the distribution tree in the inner VLAN MUST process the incoming echo request and send back echo reply.

- If Target RBridge TLV is present in the echo request:

  An RBridge owning any one of the specified target nicknames in the incoming echo request MUST send back echo reply when it is a leaf node of the distribution tree in the inner VLAN.

If echo reply has already been generated for the incoming echo request, RBridge will not generate 'Hop Count is Zero' error notification even when the hop count value in the incoming echo request is one.

Echo request with 'H' flag unset is for ping like application. It should be noted that if an RBridge receives an echo request with its own nickname listed as one of the targets, it does not send back the echo reply if the RBridge did not advertise its interest of inner
VLAN. That is to say, the connectivity check using ping in multi-
destination path is constrained by inner VLAN. Normally VLAN 1 is the
default VLAN and enabled on every RBridge. Therefore it is
recommended to put inner VLAN to be 1 when we want to check the
connectivity without the constraint of a particular customer VLAN. We
may use the echo replies from that to plot the whole distribution
tree.

7.2.2. If H (Respond Only When Hop Count is Zero) Flag Is Set

When hop count of the incoming echo request is not one, RBridge would
never generate any echo reply or 'hop count iz zero' error
notification.

If the hop count is one in the incoming echo request:

- If Target RBridge TLV is not present in the echo request:

  RBridge receiving the incoming frame with hop count equal to 1
  MUST send back error notification of 'Hop Count is Zero'. RBridges
  MUST not generate any echo reply in this case. If hop count in
  incoming echo request is more than 1, control plane will not do
  anything. RBridge forwards the frame as normal multi-destination
  TRILL frame in data plane.

- If Target RBridge TLV is present in the echo request:

  RBridges owning the only target nickname listed in TLV MUST send
  back echo reply if it is a leaf node of the inner VLAN in the
distribution tree. If it is not a leaf node of the inner vlan, no
  echo reply will be generated by the owner RBridge; however, 'Hop
  Count is Zero' error notification will be sent back instead.

  If an RBridge not owning the only target nickname listed in TLV
  receives the incoming frame with hop count equal to 1, it SHOULD
  check its LSDB. If it sits in-between of the ingress RBridge and
  the target RBridge along the specified distribution tree, RBridge
  MUST send back the error notification of 'Hop Count is Zero';
  otherwise the RBridge should not generate such error notification.
  The purpose of suppressing the error notification here is to make
  sure the ingress only receives the error notification along the
  real data path and to reduce the processing burden at ingress.

  If RBridge not owning the first target nickname listed in TLV
  receives the incoming frame with hop count greater than 1, the
  frame is forwarded as usual.
Echo request with 'H' flag set is for traceroute like applications. For traceroute with unspecified target, the ingress RBridge will be able to construct the whole distribution tree (when tree is not pruned) or the distribution tree of inner vlan (when tree is pruned by inner VLAN) according to the returned error notifications. For traceroute with a specified target in an inner VLAN, the ingress RBridge will receive the error notifications from the RBridges along the path to the target in the tree. If the target announced its interest of the inner VLAN, it will finally send back echo reply to the ingress. If the target did not announce its interest of the inner VLAN, either the target will not receive the echo request (e.g. it is located in the tree path being pruned) or the target will send back error notification of 'Hop Count is Zero' instead of echo reply.

7.3. Sending an echo reply

The inner frame header and TRILL header fields are as follows,

- Inner.MacSA: The MAC address of the RBridge generating the echo reply
- Inner.MacDA: All-Egress-RBridges
- Inner.VLAN ID: same as Inner.VLAN ID in the received echo request to which the echo reply responds
- Ingress RB Nickname: the nickname of the RBridge generating the echo reply.
- Egress RB Bridge Nickname: the ingress RBridge nickname in the corresponding received echo request
- M bit: 0
- Hop Count: defaults to the maximum value 0x3F. It can be any value that is believed to be larger than the number of hops from ingress to egress RBridge.

The values of Sender’s Instance, Sequence Number and Timestamp sent in an echo reply MUST be same as those in its corresponding echo request. H flag MUST be zero in echo reply. The value of Timestamp Received is set to the time-of-day in NTP format [NTP] according to the receiver’s clock.
If Target RBridge TLV was present in the echo request, the corresponding echo reply SHOULD copy it.

Next Hop Nickname and Incoming Port ID TLV [RBridgeOAM] MAY be included in echo reply.

When an echo reply is going to be sent to the originator RBridge, 'Hop Count is Zero' error notification MUST not be sent in response to the same echo request.

7.4. Receiving an echo reply

An RBridge SHOULD use the Sender’s Instance and Sequence Number to match up the received echo reply with the echo request it sent. If there is no match found, the RBridge should discard the echo reply.

If Jitter TLV was present in the echo request, the round trip time should not be calculated based on the difference between the arriving time of echo reply and the value of "TimeStamp sent" in the replying frame. However the single trip time is always correct to be calculated on Timestamp Received minus Timestamp Sent when the clocks of sender and receiver are synchronized.

When an RBridge receives either an echo reply or 'hop count is zero' error notification from the target RBridge for traceroute like application, it SHOULD stop sending echo request with increased hop count value.

8. Security Considerations

The security vulnerabilities raised in [RBridgeOAM] also apply to the multi-destination RBridge ping in this document. The same mechanisms can be used to prevent or alleviate the security issues.

9. IANA Considerations

New error notification sub-code needs to be allocated by IANA as specified in Section 7.

10. References

10.1. Normative References

10.2. Informative References


11. Acknowledgments

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

Authors’ Addresses

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56624558
Email: liyizhou@huawei.com

Weiguo Hao
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56623144
Email: haoweiguo@huawei.com
VLAN based Tree Selection for Multi-destination Frames

draft-yizhou-trill-tree-selection-00.txt

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on August 5, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document.

Li, et al. Expires September 5, 2012
Abstract

TRILL uses distribution trees for multi-destination traffic. In fat tree structure, it is very possible that the distance from an ingress RBridge to multiple tree roots are the same. Therefore the minimum distance comparison may not help much in the tree selection decision. Multiple trees can be used by an ingress RBridge. For any RBridge Rn, if Rn has downstream receivers of VLAN x in a distribution tree t, there will be an entry of (t, x, port list) in the multicast forwarding table on Rn. If there are n trees and m VLANs, the multicast forwarding table size on Rn is typically n*m entries. The value of m is up to 4096 and n is up to the total number of distribution trees in the campus. If finer granularity filtering such as L2/L3 multicast address is used, then the multicast forwarding table size further increases dramatically. TRILL multicast forwarding table size is limited in hardware/silicon implements and sometimes L3 multicasting shares the same table with it. This document specifies a VLAN based tree selection mechanism to reduce the multicast forwarding table size. No data plane change is required.

Table of Contents

1. Introduction ................................................ 3
2. Conventions used in this document............................ 6
3. VLAN based Tree Selection.................................... 6
   3.1. Overview ............................................... 6
   3.2. Sub-TLVs for the Router Capability TLV ................. 7
      3.2.1. The Tree Identifier and VLANs Sub-TLV ............. 7
      3.2.2. The Tree Used Identifier and VLANs Sub-TLV......... 8
   3.3. Detailed Processing..................................... 9
   3.4. Failure Handling....................................... 10
   3.5. Extensions ............................................ 10
4. Backward Compatibility...................................... 10
5. Security Considerations..................................... 12
6. IANA Considerations ...................................... 12
7. References ................................................. 12
   7.1. Normative References................................... 12
   7.2. Informative References ............................... 12
1. Introduction

One or more distribution trees can be used to distribute multi-destination frames in a TRILL campus. The RBridge having the highest tree root priority announces the total number of trees that are computed for the campus. It may also specify the ordered list of tree root nicknames that the other RBridges need to compute in the Tree Identifiers (TREE-RT-IDs) sub-TLV [RFC6326]. Every RBridge specifies the trees it wants to use in the Trees Used Identifiers (TREE-USE-IDs) sub-TLV and the VLAN it is interested in the Interested VLANs and Spanning Tree Roots (INT-VLAN) sub-TLV [RFC6326]. It is recommended that, by default, the ingress RBridge chooses the tree whose root is closest for multi-destination frames [RFC6325]. Trees Used Identifiers info is used to build the RPF table; Interested VLANs info is used for distribution tree pruning and the multicast forwarding table with pruned info is built based on that. Each distribution tree SHOULD be pruned per VLAN, eliminating branches that have no potential receivers downstream [RFC6325]. Further pruning based on L2/L3 multicast address is also possible.

It is implementation dependant that how many trees to calculate, where the tree roots are located and which tree(s) to be used by an ingress RBridge. With the increasing demand to use TRILL in data center network, there are some features we can explore for multi-destination frames in the data center use case. In order to achieve non-blocking data forwarding, a fat tree structure is often used. Figure 1 shows a typical fat tree structure based data center network. RB1&RB2 are aggregation switches and RB11 to RB14 are access switches. It is a common practice to choose the tree roots to be at the aggregation switches for more efficient traffic transportation. All the ingress RBridges which are access switches have the same distance to all the tree roots.
In the structure of figure 1, if we choose to put the tree root at RB1 and RB2, the ingress RBridge (e.g. RB11) would find more than one closest tree root (i.e. RB1 & RB2). Then an ingress RBridge has two options to select: choose one and only one as distribution tree root or use ECMP-like algorithm to balance the traffic among the multiple trees whose roots are at the same distance. For the former, single used tree per ingress RBridge, has the obvious problem of inefficient link usage. For example, if RB11 chooses the tree1 which is rooted at RB1 as the distribution tree, the link between RB11 and RB2 will never be used to ingress the multi-destination frame by RB11. For the latter, ECMP based tree selection can have a linear increase in multicast forwarding table size with the number of trees as follows.

In some implementations, a multicast forwarding table on an RBridge is used to map the key of (tree nickname + VLAN) to an index to a list of ports for multicast frame replication. The key used for mapping is simply the tree nickname when the RBridge does not prune the tree and the key could be (tree nickname + VLAN + L2/L3 multicast address) when the RBridge was programmed by control plane with L2/L3 multicast pruning information.

For any RBridge RBn, for each VLAN x, if RBn is in a distribution tree t for VLAN x, there will be an entry of (t, x, port list) in the multicast forwarding table on RBn. If there are n such trees and m such VLANs, the multicast forwarding table size on RBn is n*m entries. When 4 distribution trees are used in a TRILL campus and RBn has 4K
VLANs with downstream receivers, it consumes 16K table entries. Each entry contains a distinct combination of (tree nickname, VLAN) as the lookup key. Figure 2 left table shows an example of the multicast forwarding table with 2 distribution trees.

TRILL multicast forwarding table has a limited size in hardware implementation. In some implementations, it shares with IP multicast for a total of 16K table entries. If fine-grained label is used [TrillFGL], the number of table entries will increase dramatically.

A straightforward way to alleviate the limited table entries is not to prune the distribution tree. However it can only be used in the restricted scenarios for the following reasons,

- Unnecessary bandwidth waste for multi-destination frame. There is broadcast traffic in each VLAN, like ARP and unknown unicast. In addition, if there is huge L3 multicast traffic in some VLAN, no pruning may result in worse consequence of L3 user data unnecessarily flooded. The volume could be huge if certain application like IPTV is supported. Finer pruning like pruning based on multicast group may be desirable in this case.

- Only useful at the pure transit nodes. Edge nodes always need to maintain the multicast forwarding table with the key of (tree nickname + VLAN) since the edge node needs to decide whether to replicate the frame to local access port based on VLAN. It is very likely that edge nodes are relatively low scale switches with the smaller shared table size available.

In addition to the multicast table size concern, some silicon does not support hashing based tree nickname selection at the ingress RBridge currently. VLAN based tree selection is used instead. Control plane of ingress RBridge maps the incoming VLAN x to a tree nickname t. Then data plane will always use tree t for VLAN x multi-destination frames. Though an ingress RB may choose multiple trees to be used for load sharing, it can use one and only one tree for single VLAN.

This document describes the control plane support for VLAN based tree selection mechanism to reduce the multicast forwarding table size. It consists with the silicon implementation mentioned in the previous paragraph. Here VLAN based tree selection is a general term which also includes finer granularity case, e.g. VLAN + L2/L3 multicast group based selection.
2. Conventions used in this document

The same terminology and acronyms are used in this document as in [RFC6325].

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].

3. VLAN based Tree Selection

VLAN based tree selection can be used as a complementary distribution tree selection mechanism, especially when the multicast forwarding table size is a concern.

3.1. Overview

The tree root with the highest priority announces the tree nicknames and the VLANs allowed on each tree. Such tree-VLAN correspondence announcement can be based on static configuration or some predefined algorithm. Ingress RBridge selects the tree-VLAN correspondence it wishes to use from the list announced by the highest priority tree root. It should not transmit VLAN x frame on tree y if the highest priority tree root does not say VLAN x is allowed on tree y.

If we make sure one VLAN is allowed on one and only one tree, we can keep the number of multicast forwarding table entries on any RBridge fixed at 4K maximum (or up to 16M in case of fine grained label). For example, there are two trees in the whole campus. The highest priority tree root appoints the treel to carry VLAN 1-2000 and tree2 to carry VLAN 2001-4095. With such announcement by the highest priority tree root, every RBridge which understands the announcement will not send the VLAN 2001-4095 on treel or send the VLAN 1-2000 on tree2. Then no RBridge would need to store the entries for treel/VLAN2001-4095 or tree2/VLAN1-2000. Figure 2 shows the multicast forwarding table on an RBridge before and after we perform the VLAN based tree selection. The number of entries is reduced by a factor f, f being the number of trees used in the campus. In this example, it is reduced from 2*4095 to 4095. This affects both transit nodes and edge nodes. Data plane does not change.
Two new sub-TLVs that can be carried in the Router Capability TLV for TRILL are defined below. They can be considered as analog of finer granularity of the Tree Identifiers Sub-TLV and the Trees Used Identifiers Sub-TLV in [RFC6326].

3.2.1. The Tree Identifier and VLANs Sub-TLV

The tree identifiers and VLAN (TREE-VLANs) sub-TLV is used to announce the VLANs allowed on each tree by the IS that has the highest priority tree root. Multiple instances of this sub-TLV may be carried. Same tree nickname may occur in the multiple Tree-VLAN...
Records within the same or across multiple sub-TLVs. The sub-TLV format is as follows:

```
+------------------+
|   Type          |
+------------------+
|   Length        |
+------------------+
|  Tree-VLAN Record (1) |
|  Tree-VLAN Record (N) |
+------------------+
```

where each Tree-VLAN Record is of the form:

```
+------------------------+
|      Tree Nickname     |
+------------------------+
|   RESV |    Start.VLAN    |
+------------------------+
|   RESV |    End.VLAN     |
+------------------------+
```

- Type: Router Capability sub-TLV type, set to 20 (TREE-VLANs).
- Length: 6*n bytes, where there are n Tree-VLAN Records.
- Tree Nickname: The nickname at which a distribution tree is rooted.
- RESV: 4 bits that MUST be sent as zero and ignored on receipt.
- Start.VLAN, End.VLAN: These fields are the VLAN IDs of the allowed VLAN range on the tree, inclusive. To specify a single VLAN, the VLAN's ID appears as both the start and end VLAN.

### 3.2.2. The Tree Used Identifier and VLANs Sub-TLV

This sub-TLV has the same structure as the Tree Identifiers and VLAN sub-TLV (TREE-VLANs) specified in Section 3.2.2. The only difference
is that its sub-TLV type is set to 21 (TREE-VLAN-USE), and the Tree-VLAN record listed are those the originating IS allowes.

3.3. Detailed Processing

The highest priority tree root includes all the necessary tree related sub-TLVs defined in [RFC6326] as usual and MAY optionally include the Tree Identifier and VLANs Sub-TLV (Tree-VLANs) in its LSP. The highest priority tree root may decide that each VLAN is only allowed on one and only one tree to maximize the saving in the multicast forwarding table size.

Ingress RBridge that understands the Tree-VLANs Sub-TLV should select the tree-VLAN correspondences it wishes to use and put them in TREE-VLAN-USE sub-TLV. If there were multiple tree nicknames announced in Tree-VLANs Sub-TLV for a VLAN x, ingress RBridge must choose one of them. How to make such choice is out of the scope of this document. It may be desirable to have some fixed algorithm to make sure all ingress RBs choose the same tree for VLAN x in this case. Any single VLAN that the ingress RBridge is interested in should be related to one and only one tree ID in TREE-VLAN-USE to minimize the multicast forwarding table size on other RBridges.

When ingress RBridge tries to encapsulate a multi-destination frame for VLAN x, it should use the tree nickname that it selected previously in TREE-VLAN-USE for VLAN x.

If RBridge RBn does not perform pruning at all, it builds the multicast forwarding table exactly same as that in [RFC6325].

If RBn prunes the distribution tree based on VLANs, RBn uses the information received in TREE-VLAN-USE sub-TLV to mark the set of VLANs reachable downstream for each adjacency and for each tree it is in.

Logically, ingress RBridge that does not support VLAN based tree selection is equivalent to the one that supports it and announces all the combination pair of tree-id-used and interested-vlan as TREE-VLAN-USE.

RBn may additionally use a flag or special loopback port in a multicast forwarding table entry to indicate a multi-destination frame need to be decapsulated locally and replicate to access ports.
3.4. Failure Handling

Failure of a tree root: It is the responsibility of the highest priority tree root to inform others the change of the allowed tree-VLAN correspondence. When the highest priority tree root learns the root of tree t fails, it should re-assign the VLANs allowed on tree t to other trees.

Failure of the highest priority tree root: It is recommended to configure the second highest priority tree root with the same knowledge of the tree-VLAN correspondence allowed as that in the highest priority tree root. Once the original highest priority tree root fails, the second highest priority tree root can take over the job. The original highest priority tree root normally is also served as a distribution tree root for a bunch of VLANs. Those VLANs should be re-assigned to other tree roots by the new highest priority tree root, especially when the failed node is the only tree that allows those VLANs.

In some transient moment or misbehave of the highest priority tree root, the following errors may occur:

- No tree has been announced to allow VLAN x frames

- An ingress RBridge is supposed to transmit VLAN x frames on tree t, but root of tree t is no longer reachable.

For second case, an ingress Bridge should choose another reachable tree root which allows VLAN x by the highest priority tree root announcement. If there is no such tree available, then it is same as the first case above. Then the ingress RBridge should be ‘downgraded’ to a conventional BRbridge in [RFC6325].

3.5. Extensions

VLAN based tree selection can be easily extended to (VLAN+L2/L3 multicast group) based tree selection. For example, we can appoint multicast group 1 in VLAN 10 to treel and appoint group 2 in VLAN 10 to tree2 for better load sharing. New sub-TLVs can be specified later for this purpose.

4. Backward Compatibility

RBridge MUST always include the TREE-USE-IDs and INT-VLAN sub-TLVs as usual no matter if it supports TREE-VLAN-USE sub-TLV.
RBridge that understands TREE-VLAN-USE sub-TLV sent from another RBridge RBn should use it to build the multicast forwarding table and ignore the TREE-USE-IDs and INT-VLAN sub-TLVs sent from the same RBridge. It should be noted that TREE-USE-IDs and INT-VLAN sub-TLVs are still useful for other purposes, e.g. RPF table building, spanning tree root notification, etc. If the RBridge does not receive TREE-VLAN-USE sub-TLV from RBn, it uses the conventional way described in [RFC6325] to build the multicast forwarding table.

For example, there are two distribution trees, tree1 & tree2 in the campus. RB1&RB2 are new RBriges which use the new sub-TLVs described in this document. RB3 is an old RBridge which is compatible with [RFC6325]. RB1 receives ((tree1, VLAN10),(tree2, VLAN11)) as TREE-VLAN-USE sub-TLV and (tree1, tree2) as TREE-USE-IDs sub-TLV from RB2 on port x. RB1 also receives (tree1) as TREE-USE-IDs sub-TLV and no TREE-VLAN-USE sub-TLV from RB3 on port y. Assume RB2 is interested in VLAN 10&11 and RB3 is interested in VLAN 100&101. RB2 & RB3 announce their interested VLANs in INT-VLAN sub-TLV as usual. Then RB1 will build the entry of (tree1, VLAN10, port x) and (tree2, VLAN11, port x) based on RB2’s LSP and mechanism specified in this document. RB1 also builds entry of (tree1, VLAN100, port y), (tree1, VLAN101, port y), (tree2, VLAN100, port y), (tree2, VLAN101, port y) based on RB3’s LSP in conventional way. The complete multicast forwarding table on RB1 would be like the following.

+--------------+-----+---------+
|tree nickname |VLAN |port list|
+--------------+-----+---------+
|   tree 1     |  10 | x       |
+--------------+-----+---------+
|   tree 1     | 100 | y       |
+--------------+-----+---------+
|   tree 1     | 101 | y       |
+--------------+-----+---------+
|   tree 2     |  11 | x       |
+--------------+-----+---------+
|   tree 2     | 100 | y       |
+--------------+-----+---------+
|   tree 2     | 101 | y       |

It is expected that the table is not shrunk as small as the one where every RB supports the new TREE-VLAN-USE sub-TLVs. The worst case in a hybrid campus is the number of entries equal to the number in current
practice which does not support VLAN based tree selection. Such extreme case happens when the interested VLAN set from the new RBridges is a subset of the interested VLAN set from the old RBridges.

VLAN based tree selection is compatibility with the current practice. Its effectiveness increases with more RBridge supporting this feature in the TRILL campus.

5. Security Considerations

This document does not change the general RBridge security considerations of the TRILL base protocol. See Section 6 of [RFC6325].

6. IANA Considerations

IANA is requested to allocate the new sub-TLV type code as specified in Section 3.

7. References

7.1. Normative References


7.2. Informative References


8. Acknowledgments

Authors wish to thank Radia Perlman, Donald Eastlake, Rakesh Kumar R, Ma Liangliang for the valuable comments.

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

Authors’ Addresses

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56624558
Email: liyizhou@huawei.com

Weiguo Hao
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56623144
Email: haoweiguo@huawei.com

Somnath Chatterjee
IP Infusion,
RMZ Centennial, Block D
Doddanakundi Industrial Area,
Kundanahalli Main Road,Mahadevapura Post,
Bangalore - 560 048 Karnataka, India

Email: somnath.chatterjee01@gmail.com
TRILL IS-IS MTU Negotiation
draft-zhang-trill-mtu-negotiation-02.txt

Abstract

The IETF TRILL protocol provides least cost pair-wise layer 2 data forwarding by using IS-IS link state routing. This document defines a new link MTU size negotiation mechanism to update the TRILL documents "Routing Bridges (R Bridges): Base Protocol Specification" and "Routing Bridges (R Bridges): Adjacency".

Status of this Memo

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

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

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

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

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

Copyright and License Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.

Mingui Zhang
Expires June 15, 2012
1. Introduction
2. Issues of Link MTU Testing
3. TRILL IS-IS MTU Negotiation
4. Determining Traffic Link MTU Size
5. Security Considerations
6. IANA Considerations
7. References

Table of Contents
1. Introduction

The base TRILL protocol includes the way RBridges determine the minimum inter-RBridge link size for the whole campus (campus-wide Sz), for the proper operation of TRILL IS-IS. [RFC6327] defines the diagram of state transitions of an adjacency. If MTU testing is in effect, the "link MTU size is successfully tested (A6)" is an event causing transition between "2-way" state and "Report" state of an adjacency. RBridges use what they believe to be the campus-wide Sz to do link MTU size testing and a successfully tested link MTU size X must be not less than the value of campus-wide Sz [RFC6325].

This document analyzes the possible issues caused by the definition that link MTU size testing depends on the collection of campus-wide Sz. In order to break the global dependence on campus-wide Sz, link-wide Lz, which is the minimum acceptable inter-RBridge link MTU size for a link, is used to replace the role of campus-wide in link MTU size testing. Based on link-wide Lz, a new link MTU size testing algorithm is designed for adjacent RBridges to determine the most suitable size of their link MTU.

There are PDUs which are limited to a local link, such as CSNPs and PSNPs. These PDUs should not be confined by the campus-wide Sz. Instead, these PDUs should be formatted not greater than the value of link-wide Lz.

1.1. Content

Section 2 analyzes the issues caused by the dependence on campus-wide Sz for link MTU size testing.

Section 3 defines a new IS-IS MTU negotiation mechanism to update [RFC6325].

Section 4 describes how link traffic MTU can be determined.

1.2. Terminology

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

2. Issues of Link MTU Testing

Link MTU size testing is defined in Section 4.3.2 of [RFC6325]. If the link MTU size is smaller than campus-wide value of Sz, which is the smallest value of originatingLSPBufferSize advertised by any RBridge in its LSP (but not less than 1470)[RFC6325], the link will
not be included in the global topology. If the link MTU size $X$ of an adjacency is successfully tested ($X \geq$ campus-wide $Sz$), its state will move from "2-way" to "Report", which is defined in [RFC6327]. The link MTU size testing depends on the believed value of campus-wide $Sz$, which can be problematic. Some issues caused by this dependence are given in the following subsections.

2.1. Campus Wide Dependence

Take Figure 2.1 as an example, all the adjacencies are in report states. After RB4 leaves the campus and its LSPs age out, RB2 and RB3 find the campus-wide $Sz$ grows. They test the MTU according to the new value of campus-wide $Sz 1800$. Since RB2 and RB3 is connected by a low-end bridge whose port MTU is 1700. The test will not be successful. This adjacency has to return to 2-way state. The state of an adjacency can be determined by another remote adjacency. The stability of the campus-wide $Sz$ in such a badly configured campus can be poor resulting in maintenance problems. (In a well-configured campus all RBridges should be configured with the same originatingLSPBufferSize.)

2.2. Inconstant Consequences of Wrong Configuration

Take Figure 2.2 as an example, the originatingLSPBufferSize value of RB3 is falsely configured to be greater than its port MTU. The link
MTU testing is successful because the campus-wide Sz 1600 is smaller than the two port MTUs of the adjacency between RB2 and RB3. The adjacency will be in "Report" state. However, when RB4 leaves the campus and the campus-wide Sz is updated to 1800, the link MTU test of link RB2-RB3 cannot be successful.

Figure 2.2: Inconstant consequences of wrong configuration

3. TRILL IS-IS MTU Negotiation

In order to solve the problems depicted in Section 2, this draft introduces a new value "Lz" which is the acceptable inter-RBridge link size required by each RBridge. Link-wide Lz is the minimum Lz required by all R Bridges on a specific link. It is used in link MTU size testing to replace the role of campus-wide Sz. A successfully tested link MTU size X is not necessarily greater than the value of campus-wide Sz any more. Section 3.2 will define how to test this X value based on the value of link-wide Lz. However, the adjacency state diagram defined in [RFC6327] does not change due to the introduction of Lz. "A6" is still an event causing transition between "2-way" state and "Report" state of an adjacency.

The maximum size of PDUs exchanged only between neighbors instead of the whole campus should be confined by link-wide Lz instead of the campus-wide Sz. CSNPs and PSNPs are such kind of PDUs. They are exchanged just on the link after a DRB is selected on the link.

As for campus-wide Sz, R Bridges continue to be propagated their originatingLSPBufferSize across the campus through the advertisement of LSPs. Each RBridge should format their "campus-wide" PDUs, such as LSPs, not greater than what they believe to be the campus-wide Sz.
3.1. Determination of Link-Wide Lz

R Bridges on a LAN link should exchange their assumption on the value of "Lz" through IIH using the originatingSNPBufferSize contained in the PORT-TRILL-VER sub-TLV (see Section 6). The smallest value of the Lz collected on a link, but not less than 1470, is the link-wide Lz. It is different from the campus-wide Sz which is determined by having each RBridge in the campus advertise its own desired value of Sz in LSPs as defined in Section 4.3.1 of [RFC6325].

With IIH, an RBridge gets the value of Lz from its neighbor not later than the time when the adjacency moves to 2-way state. An RBridge should be aware of what size of PDUs will be accepted by its neighbor without exceeding its originatingSNPBufferSize.

\[
\begin{array}{c}
Lz:1800 \\
\text{++++} \\
\text{RB1}(2000)-\text{(2000)}\text{RB2} \\
\text{++++} \\
Lz:1800 \\
\text{++++} \\
\text{RB1}(2000)-\text{(1700)}\text{B1} \\
\text{++++} \\
\end{array}
\]

Figure 3.1: Link MTU has to be negotiated

However, even all RBridges on a specific LAN link have reached consensus on the value of link-wide Lz, it does not mean that these RBridges can safely exchange PDUs between each other. Take Figure 3.1 as an example. RB1, RB2 and RB3 are three RBridges on the same LAN link and their Lz is 1800, so the link-wide Lz of this LAN link is 1800. There is an intermediate bridge (say B1) between RB2 and RB3 whose port MTU size is 1700. If RB2 sends PDUs formatted in the size of 1800, it will be discarded by B1. Therefore the link MTU size has to be tested. After the link MTU size of an adjacency is successfully tested, these CSNP and PSNP PDUs will be formatted no greater than the tested link MTU size and will be safely transmitted on this link.

3.2. Link MTU Size Testing Algorithm

A link MTU size testing method given by the last paragraph of Section 4.3.2 of [RFC6325]. The following Binary Search algorithm in which link-wide Lz is used in the testing instead of campus-wide Sz.

Step 0: RB1 sends an MTU-probe padded to the size of link-wide Lz.
1) If RB1 successfully receives the MTU-ACK to the probe of the value of link-wide Lz from RB2, then link MTU size is set to the size of link-wide Lz and stop.

2) RB1 tries to send an MTU-probe padded to the size 1470.

   a) If RB1 fails to receive an MTU-ACK from RB2 after k tries (where k is a configurable parameter whose default is 3), RB1 sets the "failed minimum MTU test" flag for RB2 in RB1’s Hello and stop.

   b) Link MTU size <-- 1470, X1 <-- 1470, X2 <-- link-wide Lz, X <-- [(X1 + X2)/2] (Operation "[...]" returns the fraction-rounded-up integer.). Repeat Step 1.

Step 1: RB1 tries to send an MTU-probe padded to the size X.

1) If RB1 fails to receive an MTU-ACK from RB2 after k tries, then:

   X2 <-- X and X <-- [(X1 + X2)/2]

2) If RB1 receives an MTU-ACK to a probe of size X from RB2 then:

   link MTU size <-- X, X1 <-- X and X <-- [(X1 + X2)/2]

3) If X1 >= X2 or Step 1 has been repeated n times (where n is a configurable parameter whose default is 5), stop. Else go to Step 1.

Since the execution of the above algorithm can be resource consuming, it is recommended that the DRB takes the responsibility to do the testing. If the testing is finished and the tested link MTU size is smaller than the original link-wide Lz or the minimum Sz that has been advertised to the DRB, the DRB should send the tested link MTU size as its local originatingSNPBufferSize in IIH and originatingLSPBufferSize in LSP number zero (shorted as LSP0). This will trigger other RBridges on the link to update their link-wide Lz and campus-wide Sz to be the size of the tested link MTU. Then CSNPs, PSNPs and LSPs (including those used for LSP database synchronization) can be rightly resized and successfully exchanged on the link.

3.3. Re-determining Campus-Wide Sz

RBridges may join in or leave the campus from time to time. The campus-wide Sz can become outdated. Section 4.3.1 of [RFC6325] does not define when to re-determine the campus-wide Sz. The following suggestions are given for campus-wide Sz re-determination.
1) When a new RB whose Sz is smaller than current campus-wide Sz joins in the campus, it MUST report its Sz in an LSP which will cause other RBridges update their campus-wide Sz. The LSPs in the campus will be resized to be no greater than the new campus-wide Sz.

2) When an RB whose originatingLSPBufferSize is right at the campus-wide Sz leaves the campus, and its LSPs are purged from the remaining campus after reaching MaxAge [ISO10589]. The campus-wide Sz ought to be recomputed as well. Frequent LSP "resizing" is harmful to the stability of the whole campus, so it should be dampened. Within the two kinds of resizing actions, only the upward resizing will be dampened. When an upward resizing event happens, a timer is set (this is a configurable parameter whose default value is 300 seconds). Before this timer expires, all subsequent upward resizing will be dampened. In a well-configured campus with all RBridges configured to have the same originatingLSPBufferSize, no resizing will be necessary.

3.4. Relationship between Port MTU and Sz

When port MTU size is smaller than the local Sz of an RBridge, this port should be explicitly disabled from the TRILL campus. On the other hand, when an RBridge receives an LSP with size greater than its local Sz or the campus-wide Sz, this LSP should be normally processed rather than discarded. If an LSP is larger than the MTU size of a port over which it is to be propagated, no attempt shall be made to propagate this LSP over the port and an LSPTooLargeToPropagate alarm shall be generated [ISO10589].

3.5. LSP Synchronization

The DRB of a LAN link is elected as early as in the "Detect" state of an adjacency. The DRB begins to send out CSNP to synchronize the LSP database of the RBridges attached to this LAN link when the adjacency between this RBridge and the DRB moves to 2-way state. If a non-DRB RBridge receives this CSNP and finds that LSPx is not in its LSP database, it will send out PSNP to request LSPx from the DRB. If a non-DRB receives this CSNP and finds that LSPx is not in the LSP database of the DRB, it will also send out LSPx to the DRB.

DRB and non-DRB on a link should start to synchronize LSP database using CSNPs and PSNPs with a neighbor when the adjacency between them moves to the 2-way state [RBclr]. The CSNPs and PSNPs should be formatted in chunks of size at most the link-wide Lz. Since the link MTU size has not been tested in the 2-way state, link-wide Lz may be greater than the actual link MTU size. In that case, an CSNP or PSNP
may be discarded if its size is greater than the link MTU size. After the link MTU size is successfully tested, the adjacencies will begin to format these PDUs in the size no greater than it, therefore these PDUs will finally successfully get through.

4. Determining Traffic Link MTU Size

Campus-wide Sz and link-wide Lz are used to confine the size of TRILL IS-IS PDUs. They are different from the MTU size restricting the size of TRILL data frames. The size of TRILL data frames is restricted by the physical links and devices. It is possible that a TRILL data frame forwarded by an RBridge is greater than the campus-wide Sz or link-wide Lz.

The algorithm defined in link MTU size testing can also be used in TRILL traffic MTU size testing, only that the link-wide Lz used in that algorithm should be replaced with the port MTU of the RBridge sending MTU probes. The successfully tested size X can be advertised as an attribute of this link using MTU sub-TLV defined in [RBisisb].

Unlike RBridges, end stations do not participate the exchange of ISIS PDUs of TRILL, therefore they can not grasp the link traffic MTU size from a TRILL campus automatically. An operator may collect these values using network management tools such as TRILL ping or TraceRoute. Then the path MTU is set as the smallest tested link MTU on this path and end stations should not generate native frames that may exceed this path MTU.

5. Security Considerations

This document raises no new security issues for IS-IS.

6. IANA Considerations

The Lz value of an IS is included in PORT-TRILL-VER sub-TLV as originatingSNPBufferSize and sent in IIH (TBD) [RBisisb]. If originatingSNPBufferSize is missing from an IIH, it is assumed that its originating IS is implicitly advertising its Lz value as 1470 octets.

7. References

7.1. Normative References


7.2. Informative References

Author’s Addresses

Mingui Zhang
Huawei Technologies Co., Ltd
Huawei Building, No.156 Beiqing Rd.
Z-park, Shi-Chuang-Ke-Ji-Shi-Fan-Yuan, Hai-Dian District,
Beijing 100095 P.R. China
Email: zhangmingui@huawei.com

Xudong Zhang
Huawei Technologies Co., Ltd
Huawei Building, No.156 Beiqing Rd.
Z-park, Shi-Chuang-Ke-Ji-Shi-Fan-Yuan, Hai-Dian District,
Beijing 100095 P.R. China
Email: zhangxudong@huawei.com

Donald E. Eastlake, 3rd
Huawei Technologies
155 Beaver Street
Milford, MA 01757 USA
Phone: +1-508-333-2270
EMail: d3e3e3@gmail.com