

TRILL Working Group  
Internet Draft  
Intended status: Standards Track

Tissa Senevirathne  
Dinesh G Dutt  
CISCO  
Vishwas Manral  
HP Networking  
Sam Aldrin  
HuaWei

July 6, 2012

Expires: January 2013

ICMP based OAM Solution for TRILL  
draft-tissa-trill-oam-04.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 December 6, 2012.

#### Copyright Notice

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

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

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

## Abstract

This document presents a solution suite for TRILL data plane monitoring and failure detection. Methods presented herein allow in-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

<a href="#">1.</a>	<a href="#">Introduction.....</a>	<a href="#">4</a>
<a href="#">1.1.</a>	<a href="#">Motivation.....</a>	<a href="#">6</a>
<a href="#">1.2.</a>	<a href="#">Contributors.....</a>	<a href="#">7</a>
<a href="#">2.</a>	<a href="#">Conventions used in this document.....</a>	<a href="#">7</a>
<a href="#">3.</a>	<a href="#">Protocol Architecture Overview.....</a>	<a href="#">7</a>
<a href="#">3.1.</a>	<a href="#">Overview of Tools.....</a>	<a href="#">8</a>
<a href="#">3.2.</a>	<a href="#">TRILL Data Plane.....</a>	<a href="#">9</a>
<a href="#">3.3.</a>	<a href="#">Monitoring.....</a>	<a href="#">10</a>
<a href="#">3.4.</a>	<a href="#">Traffic Triggered Monitoring (TTM).....</a>	<a href="#">10</a>
<a href="#">3.5.</a>	<a href="#">Distribution.....</a>	<a href="#">10</a>
<a href="#">3.6.</a>	<a href="#">ISIS.....</a>	<a href="#">11</a>
<a href="#">3.7.</a>	<a href="#">Reporting.....</a>	<a href="#">11</a>
<a href="#">4.</a>	<a href="#">Frame Format.....</a>	<a href="#">11</a>
<a href="#">4.1.</a>	<a href="#">Encoding of Request message.....</a>	<a href="#">12</a>
<a href="#">4.2.</a>	<a href="#">Encoding of Response Message.....</a>	<a href="#">13</a>
<a href="#">4.3.</a>	<a href="#">Encoding of Notification Message.....</a>	<a href="#">13</a>
<a href="#">4.3.1.</a>	<a href="#">Pseudo IP Header.....</a>	<a href="#">15</a>

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
9.2.	Loopback Message Hop-count method.....	60
9.2.1.	Identification of OAM frames.....	60
9.2.2.	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

<a href="#">9.6.</a>	<a href="#">Address-Binding Verification Message.....</a>	<a href="#">71</a>
<a href="#">9.6.1.</a>	<a href="#">Extension to ARP and invARP.....</a>	<a href="#">72</a>
<a href="#">9.6.1.1.</a>	<a href="#">Encoding ARP-invARP extensions.....</a>	<a href="#">74</a>
<a href="#">9.7.</a>	<a href="#">End-Station Attachment Point Discovery.....</a>	<a href="#">76</a>
<a href="#">9.8.</a>	<a href="#">DRB and AF Discovery.....</a>	<a href="#">77</a>
<a href="#">9.8.1.</a>	<a href="#">Theory of Operation.....</a>	<a href="#">78</a>
<a href="#">9.8.1.1.</a>	<a href="#">Originator RBridge.....</a>	<a href="#">78</a>
<a href="#">9.8.1.2.</a>	<a href="#">Receiving RBridge.....</a>	<a href="#">78</a>
<a href="#">9.9.</a>	<a href="#">Diagnostic Payload Discovery for ECMP coverage.....</a>	<a href="#">80</a>

Senevirathne

Expires January 6, 2013

[Page 3]

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

<a href="#">9.9.1.</a>	<a href="#">Theory of Operations.....</a>	<a href="#">82</a>
<a href="#">9.9.1.1.</a>	<a href="#">Receiving RBridge.....</a>	<a href="#">83</a>
<a href="#">9.10.</a>	<a href="#">Notification Messages.....</a>	<a href="#">84</a>
<a href="#">10.</a>	<a href="#">Monitoring and Reporting.....</a>	<a href="#">85</a>
<a href="#">10.1.</a>	<a href="#">Data categories.....</a>	<a href="#">87</a>
<a href="#">10.2.</a>	<a href="#">Advertising Policy.....</a>	<a href="#">87</a>
<a href="#">10.2.1.</a>	<a href="#">Multi Instance ISIS and Flooding Scope.....</a>	<a href="#">89</a>
<a href="#">10.3.</a>	<a href="#">Summary Category.....</a>	<a href="#">89</a>
<a href="#">10.4.</a>	<a href="#">Detail Category.....</a>	<a href="#">91</a>
<a href="#">10.5.</a>	<a href="#">Vendor Specific Category.....</a>	<a href="#">97</a>
<a href="#">11.</a>	<a href="#">Traffic Triggered Monitoring (TTM).....</a>	<a href="#">98</a>
<a href="#">11.1.</a>	<a href="#">TTM Policy.....</a>	<a href="#">100</a>
<a href="#">11.2.</a>	<a href="#">TTM Commands.....</a>	<a href="#">102</a>
<a href="#">11.3.</a>	<a href="#">Reverse Flow Monitoring (RFM).....</a>	<a href="#">103</a>
<a href="#">12.</a>	<a href="#">Security Considerations.....</a>	<a href="#">103</a>
<a href="#">13.</a>	<a href="#">IANA Considerations.....</a>	<a href="#">103</a>
<a href="#">13.1.</a>	<a href="#">IANA considerations.....</a>	<a href="#">103</a>
<a href="#">13.1.1.</a>	<a href="#">ICMP Extensions.....</a>	<a href="#">103</a>
<a href="#">13.1.2.</a>	<a href="#">TRILL-OAM UDP port.....</a>	<a href="#">103</a>
<a href="#">13.1.3.</a>	<a href="#">ARP Extensions.....</a>	<a href="#">103</a>
<a href="#">13.1.4.</a>	<a href="#">Well known Multicast MAC.....</a>	<a href="#">104</a>
<a href="#">13.2.</a>	<a href="#">IEEE Registration Authority Consideration.....</a>	<a href="#">104</a>
<a href="#">14.</a>	<a href="#">References.....</a>	<a href="#">104</a>
<a href="#">14.1.</a>	<a href="#">Normative References.....</a>	<a href="#">104</a>
<a href="#">14.2.</a>	<a href="#">Informative References.....</a>	<a href="#">105</a>
<a href="#">15.</a>	<a href="#">Acknowledgments.....</a>	<a href="#">105</a>
<a href="#">Appendix A.</a>	<a href="#">Reports.....</a>	<a href="#">106</a>
<a href="#">A.1.</a>	<a href="#">Sample Reports.....</a>	<a href="#">106</a>
<a href="#">A.2.</a>	<a href="#">Summary Report.....</a>	<a href="#">106</a>
<a href="#">A.3.</a>	<a href="#">Detail Report.....</a>	<a href="#">107</a>
<a href="#">A.4.</a>	<a href="#">C-Type usage in messages.....</a>	<a href="#">108</a>
<a href="#">Authors' Addresses.....</a>		<a href="#">109</a>

## 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.

Senevirathne

Expires January 6, 2013

[Page 4]

---

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

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 RBridges 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 RBridge 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 R Bridge (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 addressees the problem space.

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

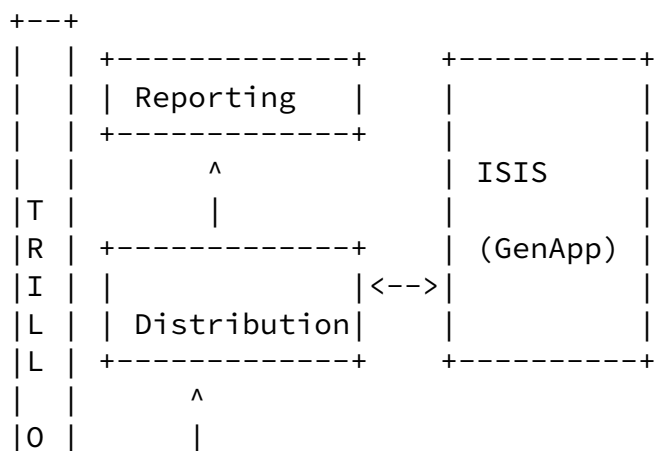
## 2. Conventions used in this document

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.





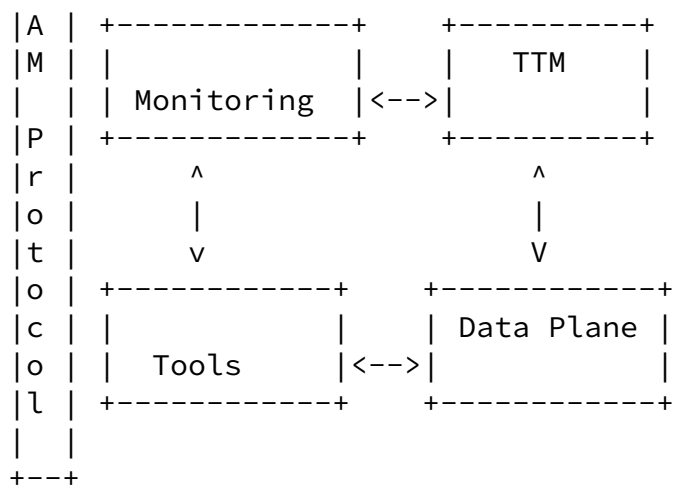


Figure 1 Architecture Overview

### 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 in to 3 broader categories as below.

Category	Tools
Fault Verification	Loop Back Message
Fault Isolation	Path Trace Message, Multicast Tree Verification
Auxiliary	MAC discovery Address Binding Verification IP End-station Locator DRB-AF Discovery Error Notification OAM command messages

### [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

- o Data from the Monitoring Layer
- o 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

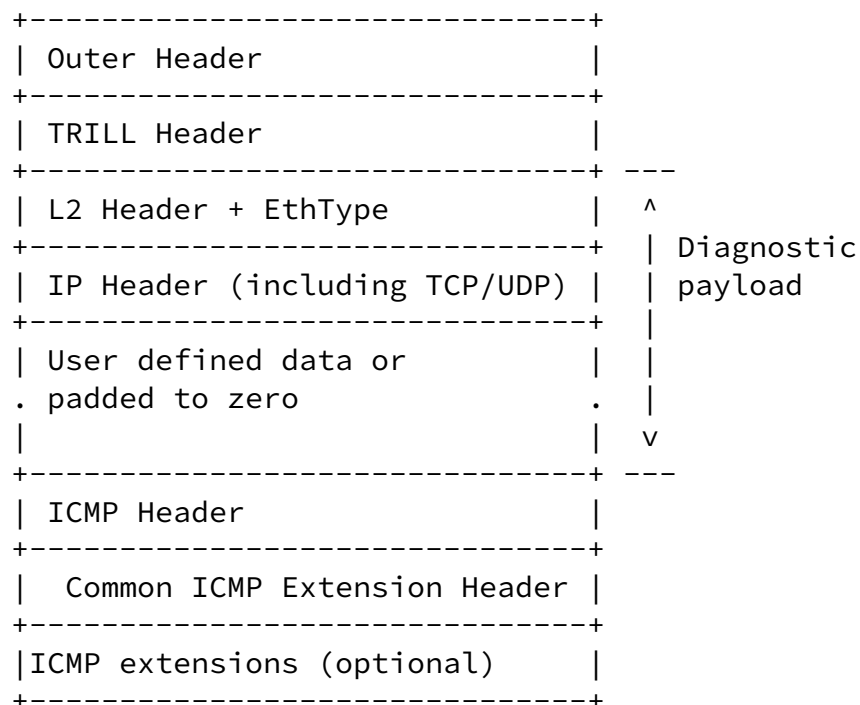


Figure 2 Encoding TRILL data plane diagnostic request message

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 allow user to specify different kinds of payloads, including actual payloads. Most hardware implementations use IPDA:IPSA:DestPort:SrcPort based hash method to select ECMP paths for IP frames. For non IP payloads, RBridges normally uses a Layer 2 MAC DA and SA based hash for selecting an ECMP path. Flexible diagnostic payload allow user to drive end to end ECMP selection based on payload without needing additional hardware. Also, in terms of forwarding, this keeps diagnostic frame 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

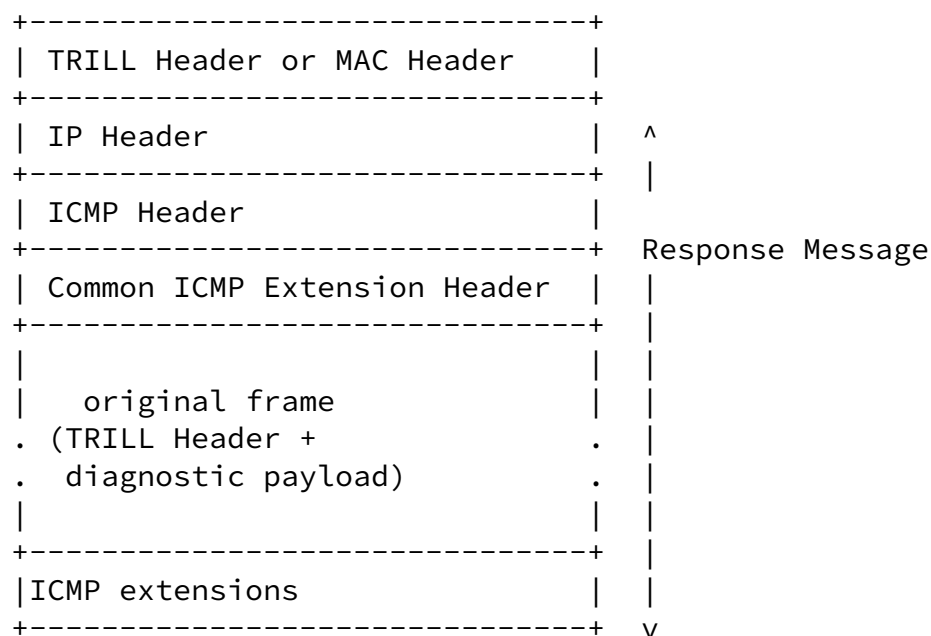


Figure 3 Encoding of OAM response message

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 128bytes 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.

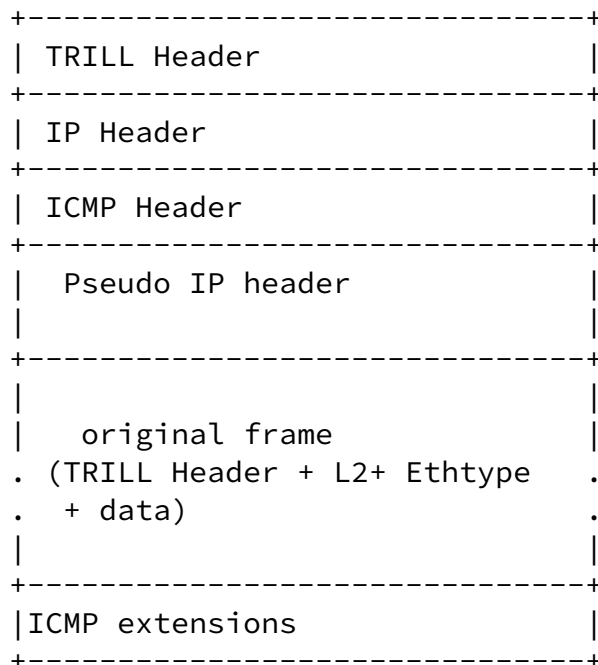


Figure 4 Encoding of Notification message

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

- o Time Expiry
- o Destination Unreachable
- o Parameter Problem

Senevirathne

Expires January 6, 2013

[Page 14]

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

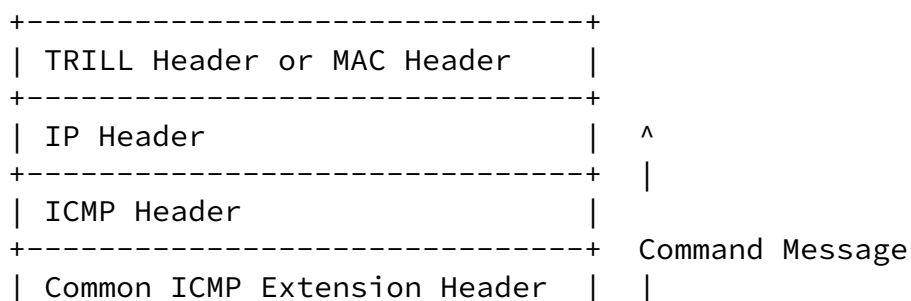
Additional TRILL OAM error codes may be specified as ICMP multipart extensions in above notifications messages. These error codes indicate the cause of the error. Please see [section 8](#). for error code definitions and [section 9.10](#). for theory of operation.

#### [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





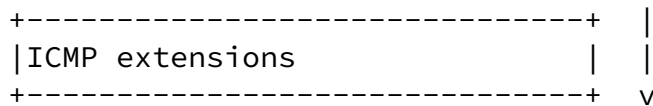


Figure 5 Encoding of OAM Command Message

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. RBridges are not required to have IP interfaces enabled. However, in order to receive and process ICMP messages, RBridges 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 RBridges 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 indentify OAM (diagnostic) frames that contain non IP diagnostic payloads.

Each RBridge, if TRILL data plane OAM enabled, MUST provide following processing:

- o 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.
- o 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:

- o 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.
- o 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:

- o 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.
- o Each RBridge SHOULD NOT egress frames with Well known MAC address as source address to non TRILL interfaces.
- o 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:

- o 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).
- o 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:

- o Overall Tree Verification Messages
- o 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 trees 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.

RBridges 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.

Flow type	Default Values	Fixed fields
Layer 2	DA= Well Known MAC SA= RBridge Interface MAC VLAN= native VLAN	EthType=OAM(TBD)
IPv4 OR IPv6	IP Address = in-band address IP Dest. Port = 3503 IP Src. Port = TBD DA = OAM MAC of egress RBridge SA = ingress RBr interface MAC VLAN= native VLAN	EthType=0x8000 OR EthType=0x86DD
Multicast Tree Verification	SA= RBridge Interface MAC VLAN= native VLAN	DA= Well Known Multicast MAC EthType=OAM(TBD)
Layer 2 Multicast	DA= Well Known MAC SA= RBridge Interface MAC VLAN= native VLAN	EthType=OAM(TBD)
IP Multicast	IP Dest Address = Default OAM MCast address IP Src. Address = in-band-address IP Dest. Port = 3503 IP Src. Port = TBD DA = OAM MAC of egress RBridge SA = ingress RBr interface MAC VLAN= native VLAN	EthType=0x8000 OR EthType=0x86DD

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].

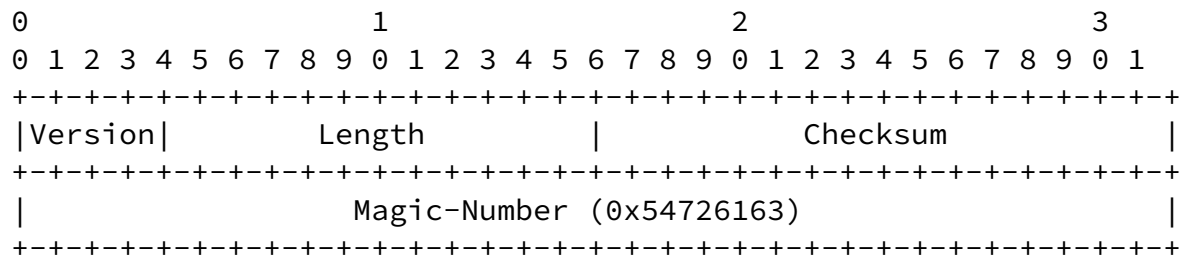


Figure 7 ICMP Common Extension Header

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:

- o OAM capability
- o OAM in-band IP address
- o 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.

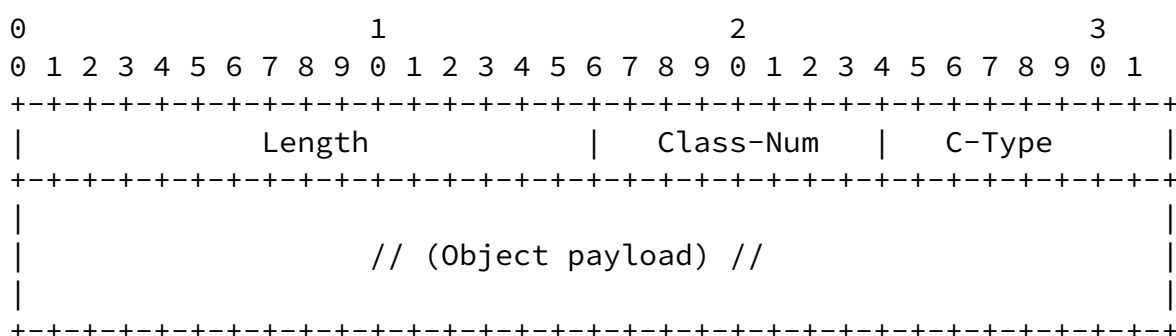


Figure 9 ICMP Extension Object

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.

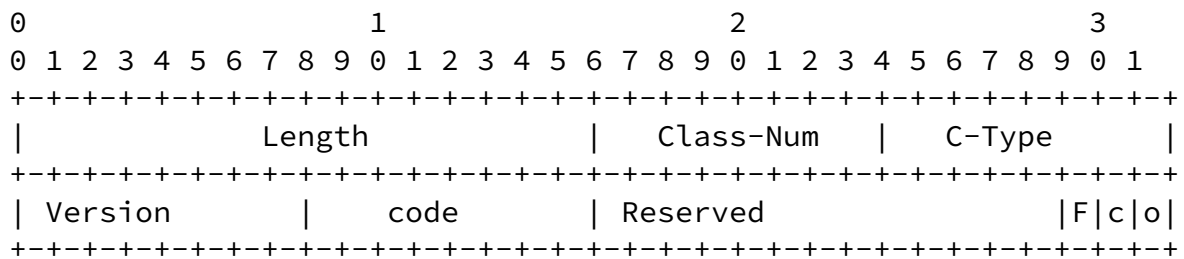


Figure 10 Example of Encoding Version and Flags object

#### 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.

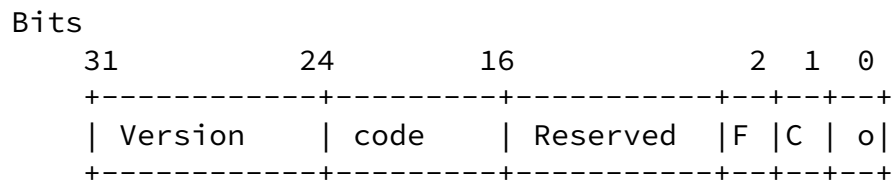


Figure 11 C-Type 1, Version and Flags

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.

0 (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
- 25 : Loopback Message response for message code 24.
- 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

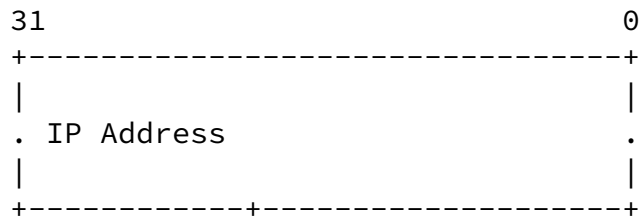


Figure 12 C-Type 2 Originator IP address

#### 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

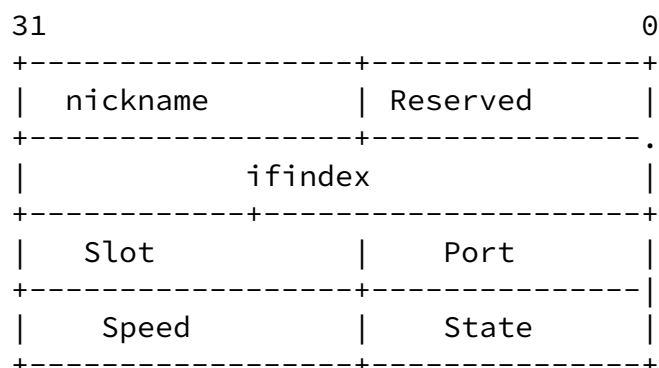


Figure 13 C-Type 3 Upstream Identification

Nickname (2 octets): TRILL 16 bit nickname of the upstream RBRdige.  
[RFCtrill]

Reserved (2 octetc) : 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.

Bits

16 0  
+-----+



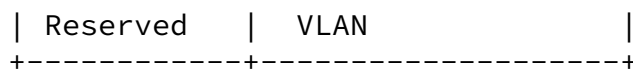


Figure 14 C-Type 4 Monitored (Diagnostic) VLAN

#### 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.

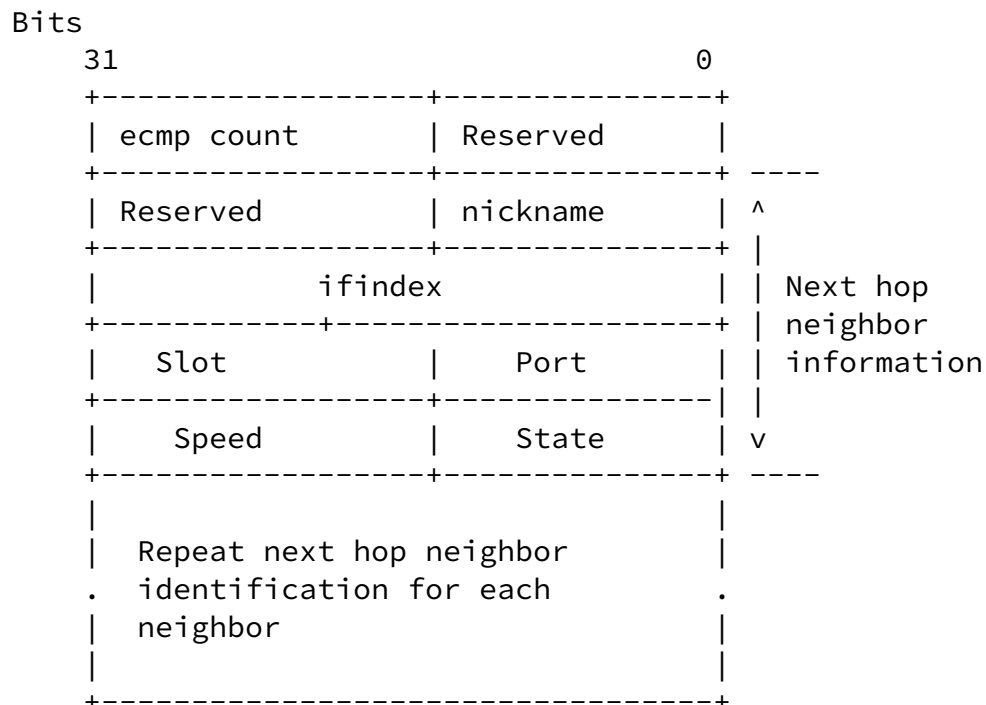


Figure 15 C-Type 5 Downstream Identification

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.

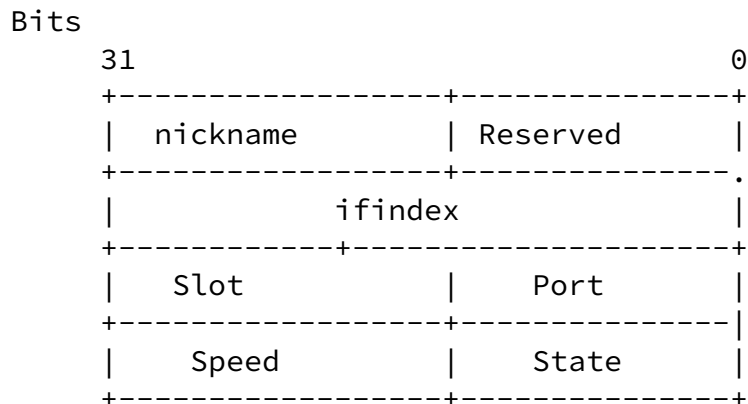


Figure 16 C-Type 6 Path for this payload

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)

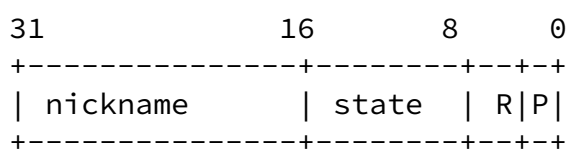


Figure 17 Nickname of the DRB

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

Nickname (2 octets) : TRILL nickname of the DRB

State (1 octets) : DRB state

R (7 bits) : set to zero on Transmission and ignored on receipt

P (1 bits) : 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)

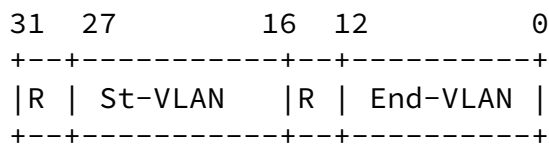


Figure 19 Announcing VLAN List

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 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.

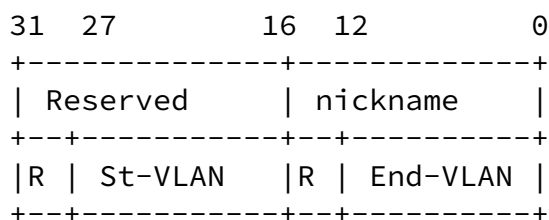


Figure 20 AF List

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.

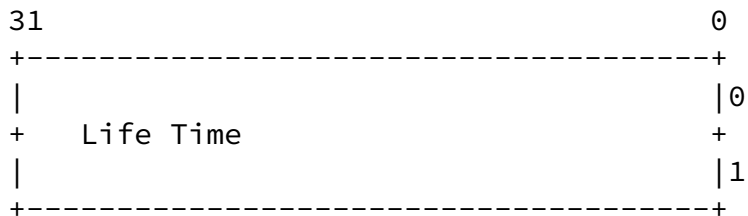


Figure 21 DRB Life Time

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 R Bridges.

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 R Bridges, where applicable.

Multicast Tree downstream list provides information on downstream leaf Rbridges on the specified tree.

Bits

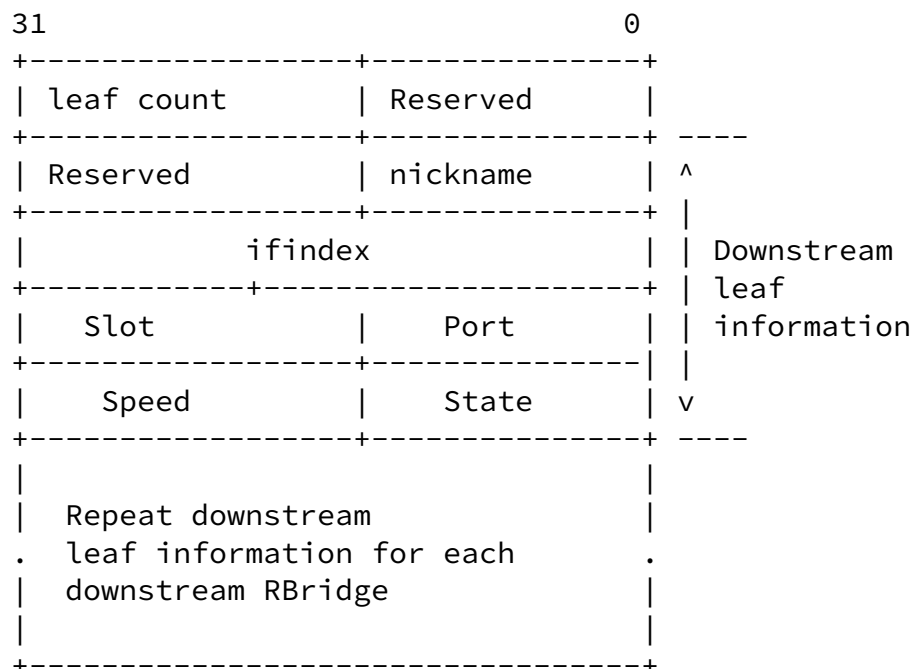




Figure 24 C-Type 15 Multicast Tree Downstream List

Leaf count (16 bits): Number of RBridges downstream to this RBridge.

Downstream leaf information:

Nickname (16 bits): TRILL 16 bit nickname [RFCtrill]

Ifindex (32 bits) : Unsigned 32 bit integer that has only a local significance to the sending RBridge. 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.

Senevirathne

Expires January 6, 2013

[Page 37]

---

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

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)

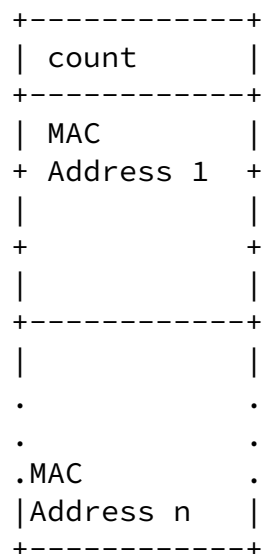
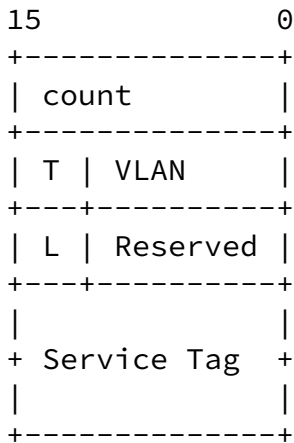


Figure 25 MAC-discovery Address List

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)



+ MAC	+
Address	
+	+
+-----+	
Age	
+	+
+	+
+	+
+-----+	
Ifindex	
+	+
+-----+	
vNTAG	
+-----+	
Slot	
+-----+	
Port	
+-----+	
State	
+-----+	
Speed	
+-----+	

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 octes): 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.

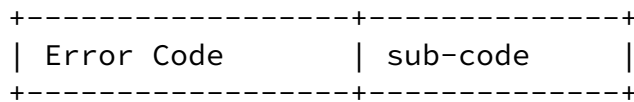


Figure 27 C-Type 18 Error code

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 - Unknwon 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.

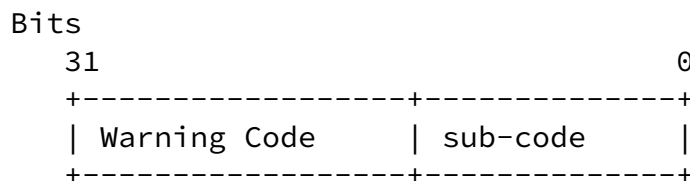


Figure 28 C-Type 19 Warning code

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.

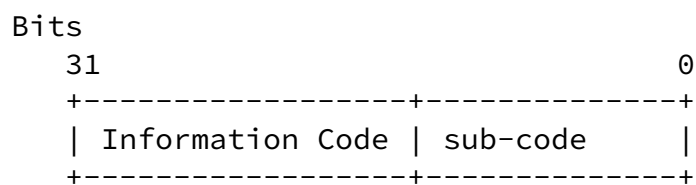


Figure 29 C-Type 20 Information code

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](#)]).

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

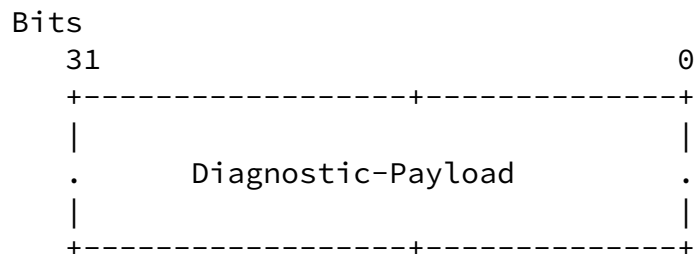


Figure 30 C-Type 21 Information code

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 to 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.

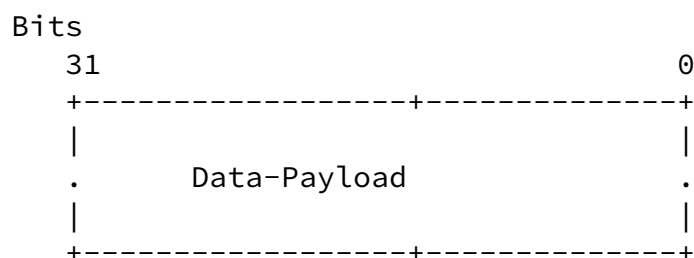


Figure 31 C-Type 21 Information code

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.

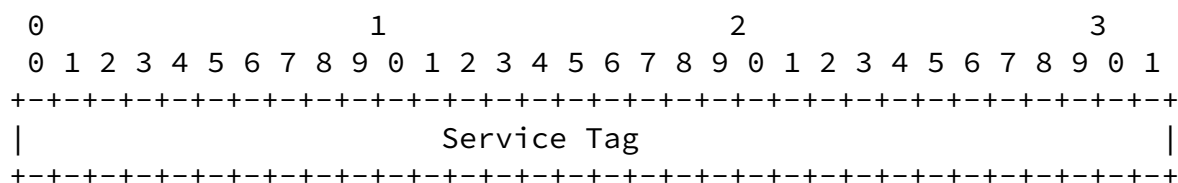


Figure 32 C-Type 23 Service Tag

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.

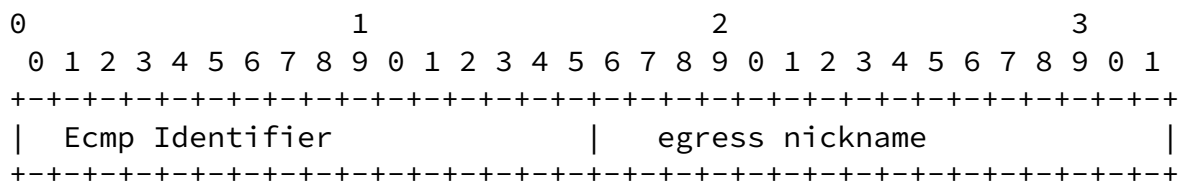


Figure 33 C-Type 24 control Plane Forwarding Verification Request

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.

[Page 44]

July 2012

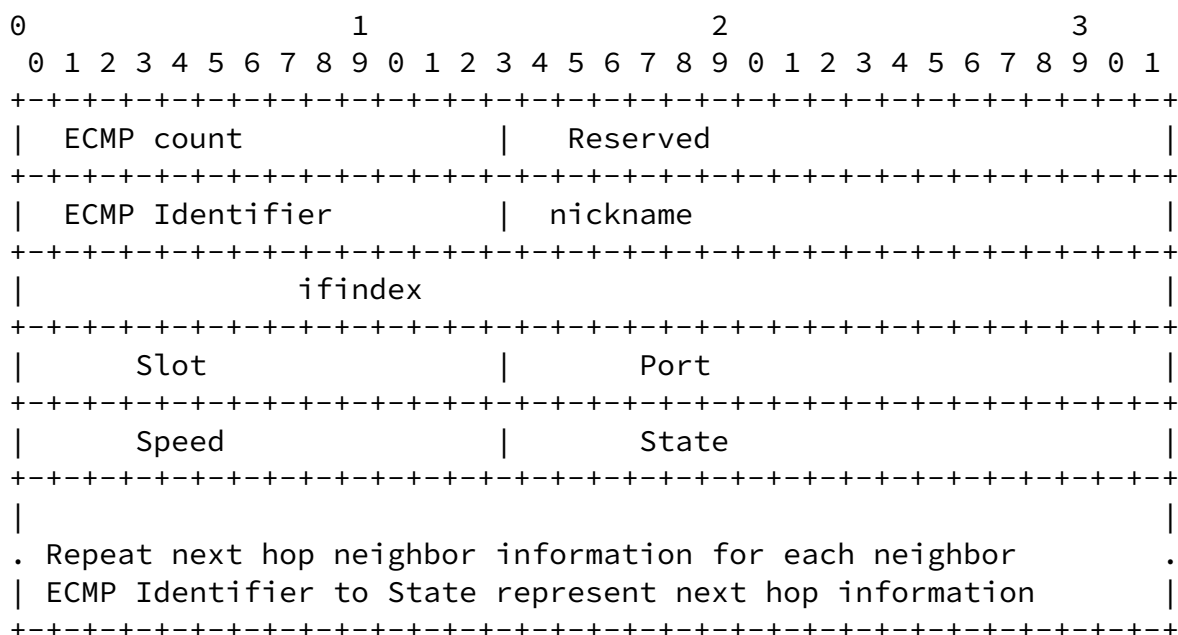


Figure 34 C-Type 25 control Plane Forwarding Verification Response

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

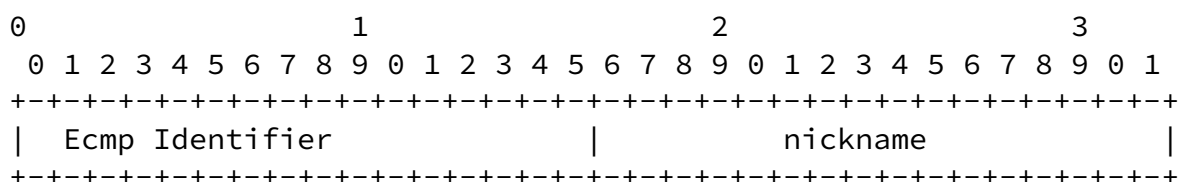


Figure 35 C-Type 26 Reverse Path Forwarding Verification Request

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.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Ecmp count                |  Reserved                        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  ECMP Identifier            |  nickname                        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               ifindex                          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Slot                |          Port                    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Speed                |          State                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |                               |
. Repeat next hop neighbor information for each neighbor        .
| ECMP Identifier to State represent next hop information        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 36 C-Type 27 Reverse Path 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.

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.

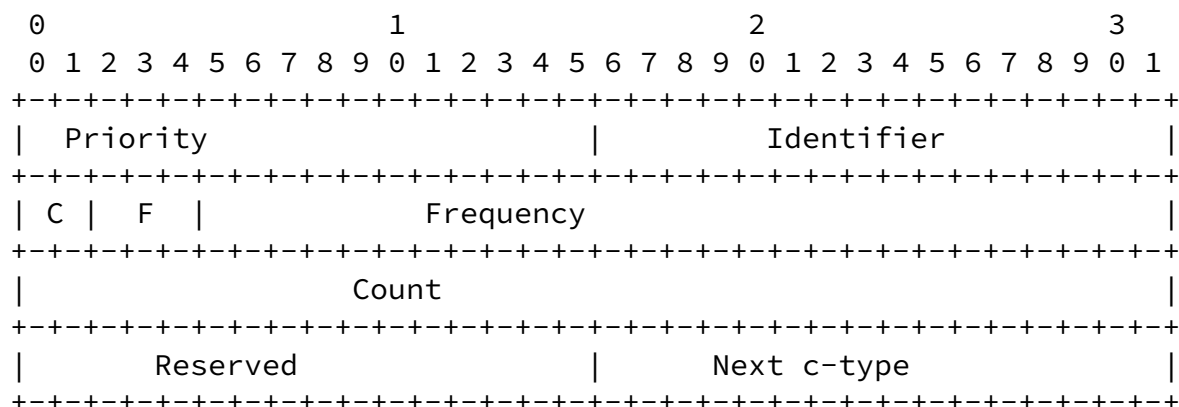


Figure 37 C-Type 28 TTM Profile

C Indicate the Class

TTM Profile action (c-Type 29)

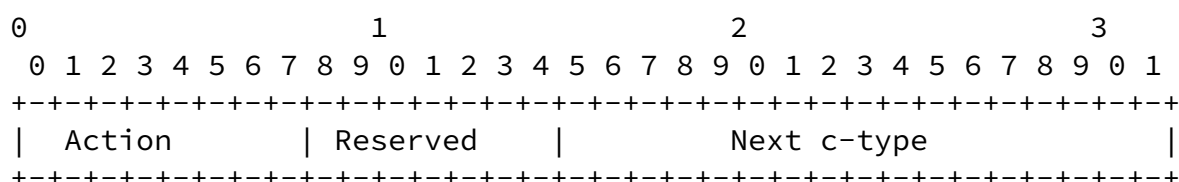


Figure 38 C-Type 29 TTM action

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)

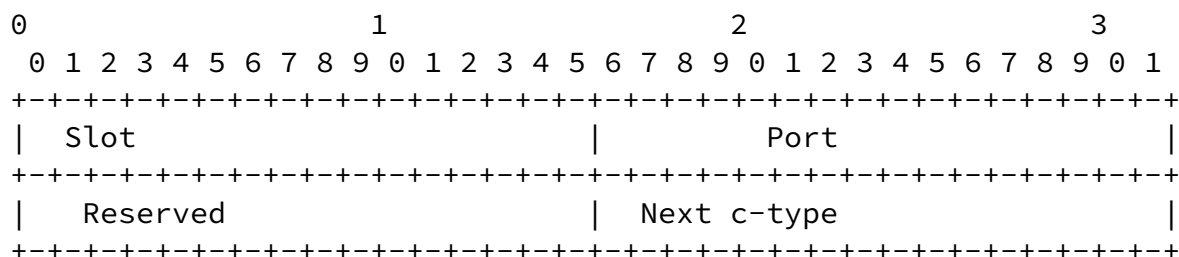


Figure 39 C-Type 30 TTM Test Point

NOTE: Given TTM Profile may contain multiple Test Points.

#### TTM Ingress End Point (c-type 31)

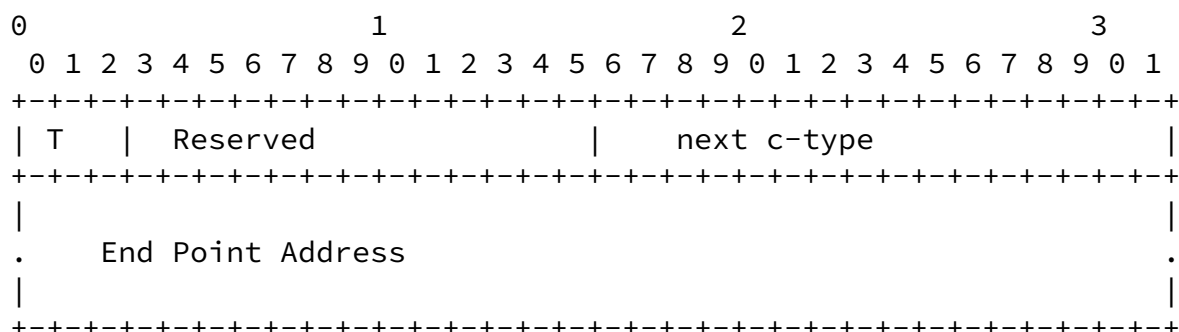


Figure 40 C-Type 31 TTM Ingress End Point

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)

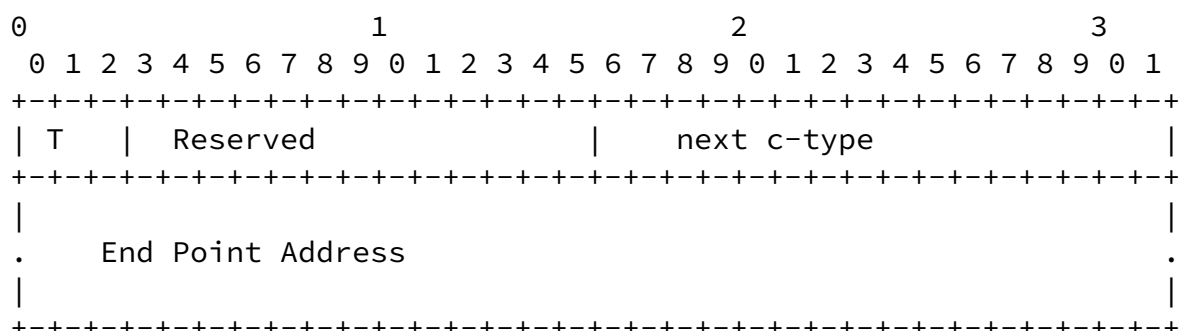


Figure 41 C-Type 32 TTM Egress End Point

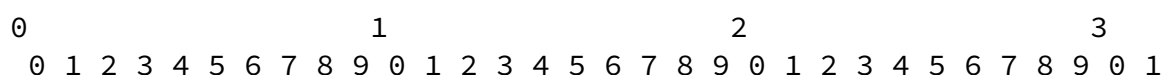
T 3 bits:

- 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.

#### TTM Pattern (c-type 33)



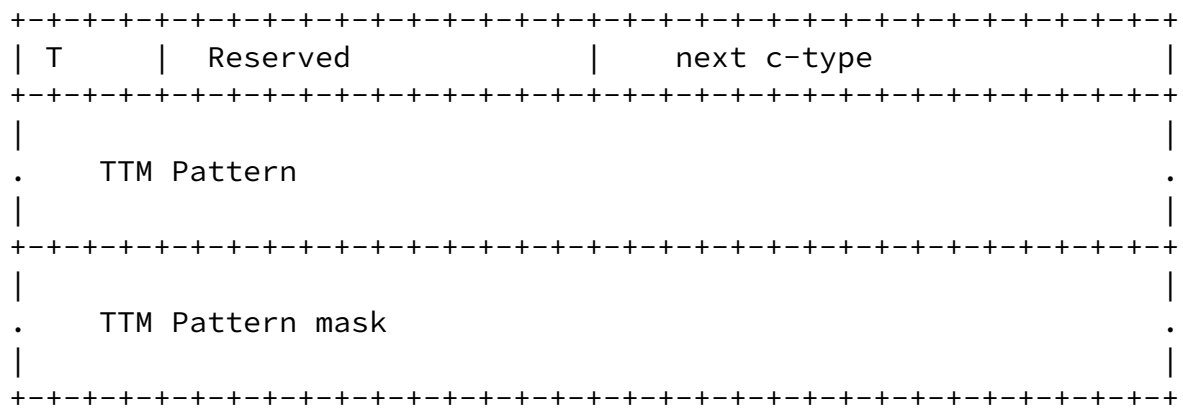


Figure 42 C-Type 33 TTM Pattern

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 do not 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)



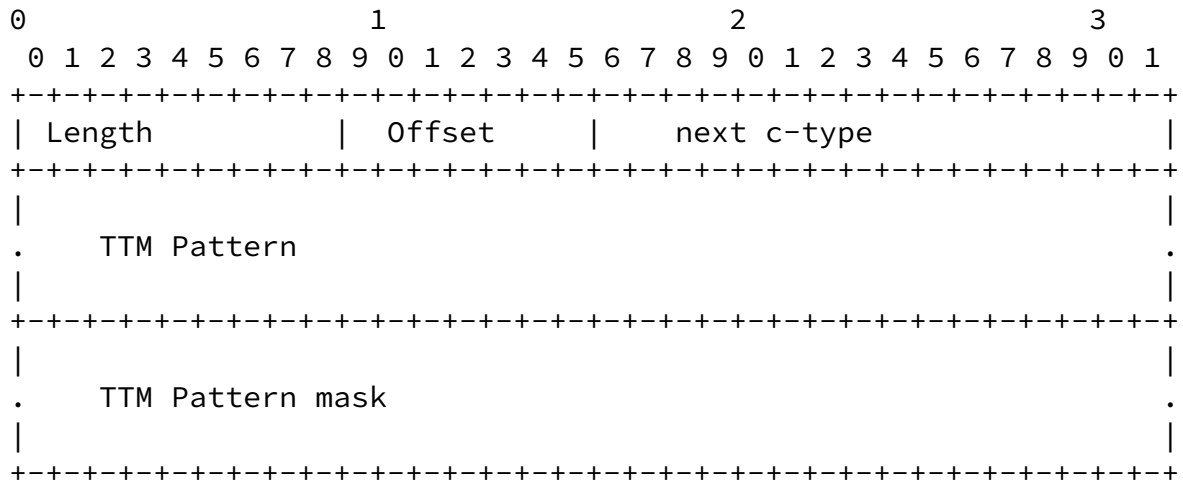


Figure 43 C-Type 34 TTM Opaque pattern

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.

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

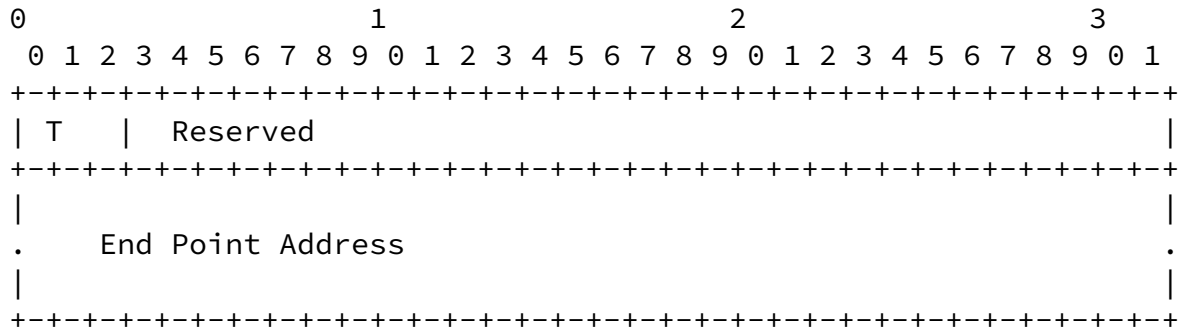


Figure 44 C-Type 35 End Point

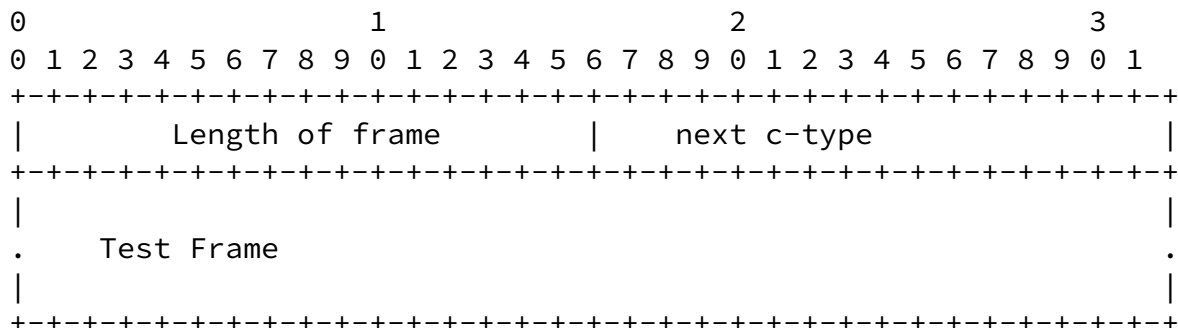
T 3 bits:

- 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.



Length (2 octets) specify the length of the Test Frame

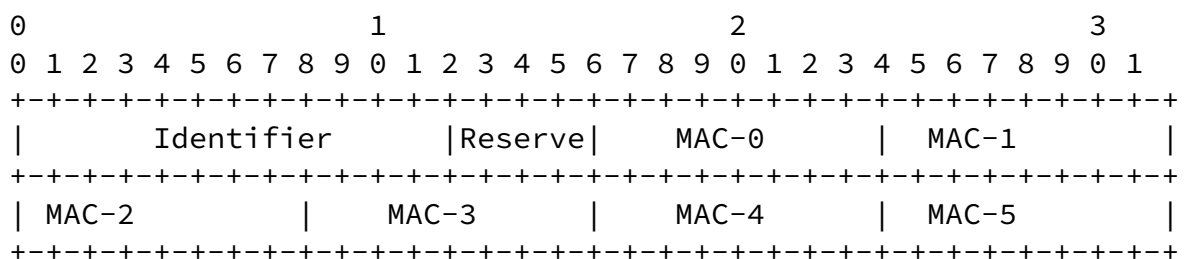
Next c-type is the next c-type within the TTM profile.

[Page 53]

July 2012

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 in to 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.



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)

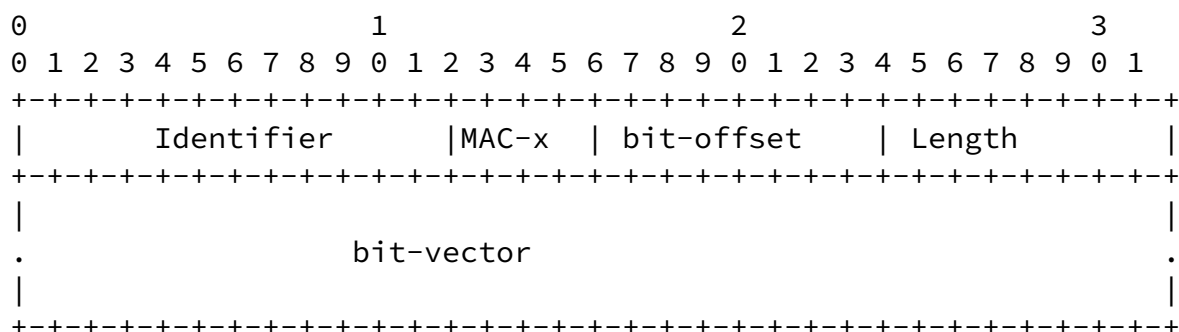


Figure 47 C-Type 39 MAC-Octet bit vector

Identifier (12 bits), uniquely identifies a MAC address seed within a diagnostic payload discovery message.

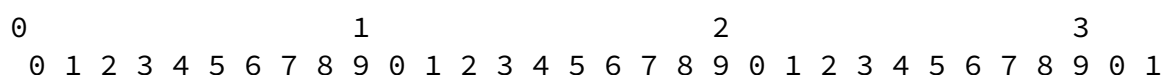
MAC-x : Value 0-5 indicates the MAC address octet represented by this bit-vector.

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)





| Sub-code | Reserved | status |
|----------|----------|--------|
| 0        | 1        | 2      |

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)

| 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 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 | 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 |  |  |  |  |  |  |          |  |  |  |  |  |  |  |  |  |
| Reserved                                    |                                             |                                             |                                             |  |  |  |  |  |  | Eth Type |  |  |  |  |  |  |  |  |  |

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

- o Out-of-band response request
- o Out-of-band IP address
- o 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

- o Out-of-band response request
- o Out-of-band IP address
- o Diagnostic VLAN

Specify the Hop Count of the TRILL data frame as 1 for the first frame. Or use Hop 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 Hop 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 Hop 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 Hop 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

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

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

- o Out-of-band response request
- o Out-of-band IP address
- o Diagnostic VLAN
- o In scope RBridge list.
  - o 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](#). .

Senevirathne

Expires January 6, 2013

[Page 65]

---

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

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 [section 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 wheather 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

- o RBRidge nickname where the MAC address is learnt
- o Interface Index and Name on which the MAC address is learnt
- o Type (i.e. Static, Dynamic, Secure etc.)
- o Age of the MAC address
- o Virtual Interface Tag (vNTAG)
- o Interface Type (Legacy or TRILL Shared)

- o DRB on the VLAN (If Applicable)
- o AF for the VLAN (If Applicable)
- o Time AF operational (If Applicable)

Optionally, an implementation may include the following information

- o System MAC address of the device connected to the port with which the MAC address is associated.
- o System information, such as name, IP address and location of the device connected to the port with which the MAC address is associated.
- o 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 EthtType. 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

- o Version
- o Set the OAM code to the MAC address discovery request (9)
- o Indicate that this is a MAC discovery message
- o One or more MAC address to be discovered
- o VLAN ID of MAC addresses (optional)
- o Service Tag that represent the overlay network (optional)
- o Out-of-band response request (optional)
- o Out-of-band IP address (optional)
- o 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 unciast.

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 payload 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)

- o Time AF operational (If Applicable)

Optionally an implementation may include the following information:

- o The system MAC address of the device connected to the port with which the MAC address is associated.
- o System information, such as name, IP address and location of the device connected to the port on which MAC address is associated.
- o 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 interface with the physical server.

It is important to have 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 in to 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.



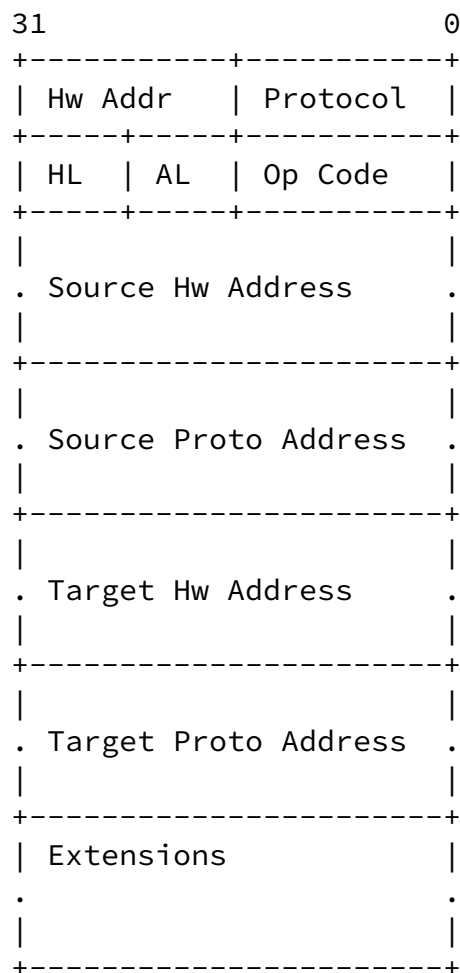


Figure 52 Encoding of ARP and invARP

#### [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.

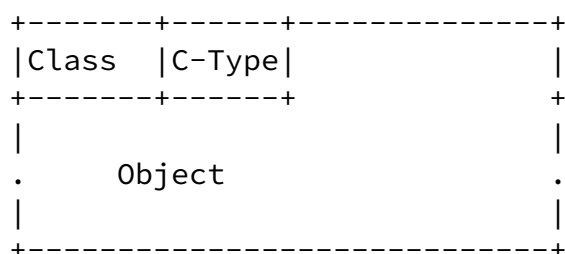


Figure 53 Encoding of ARP Extensions

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)

| Class | C-Type | Name    | Description                                       |
|-------|--------|---------|---------------------------------------------------|
| 1     | 1      | vNTAG   | vNTAG of the interface                            |
| 2     | 1      | RBridge | TRILL RBridge nickname                            |
|       | 2      | ifindex | ifindex of RBridge interface ARP response arrived |
|       | 3      | Slot    | Slot id of RBridge interface ARP response arrived |
|       | 4      | Port    | Port id of RBridge interface ARP response arrived |



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:

- o Nickname of the DRB
- o STP Root Bridge identifier
- o Up time of AF (if responder is the AF)
- o Up time of DRB (if Responder is DRB)
- o Enabled VLAN List

Senevirathne

Expires January 6, 2013

[Page 77]

---

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

- 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:

- o Response code set AF discover response (14) or AF-VLAN discover response (16)
- o Nickname of the DRB
- o Nickname of AF of the specified VLAN or AF VLAN-List if VLAN is not specified.
- o STP Root Bridge identifier
- o AF Life time (i.e. How long has been AF)
- o Enabled VLAN List
- o Announcing VLAN List
- o AF State
- o 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 discover 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 (RB*i*) 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 (RB*j*) downstream from RB*i* 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
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+
|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 1 0 1 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
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+
|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 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0 0 0 0 1 1 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|1 1 1 1|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

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.

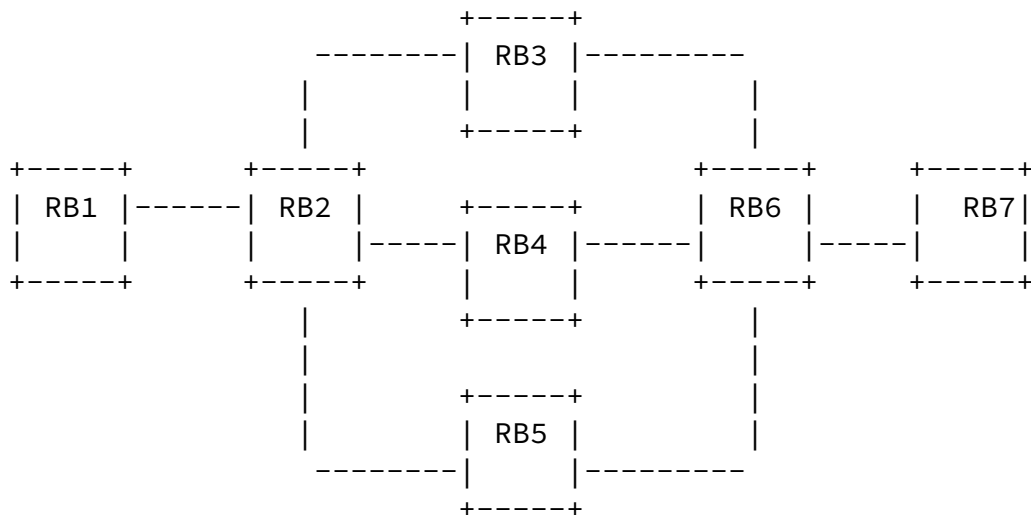


Figure 55 Sample Topology

#### [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

- o Payload Generation Request (c-type 40)
  - o Egress RBridge nickname (c-type 40)
  - o ECMP choices (c-type 40)
- o Interested VLAN (c-type 4)
- o Interested EthType (c-type 43)
- o Seed Destination MAC Address (c-type 37)
- o Seed Source MAC Address (c-type 38) (optional)
- o MAC Octet bit-vector (c-type 39)
- o Service Tag (c-type 23) (optional)

#### TLVS in the Command Response Message

- o Payload generation response (c-type 41) (for each ECMP choice)
  - o ECMP choice status (for each of the requested ECMP)
- o Interested VLAN (c-type 4)
- o Interested EthType (c-type 43)
- o Seed Destination MAC Address (c-type 37)
- o Seed Source MAC Address (c-type 38) (optional, included only if present in the request)
- o MAC Octet bit-vector (c-type 39) (bit values appropriately set)
- o 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:

- o Time Expiry
- o Destination Unreachable
- o 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.

- o Egress RBridge nickname unknown
- o Inner VLAN does not exist or suspended
- o Not the AF for inner VLAN

Parameter Problem notification may be generated for following scenarios; additional scenarios may be added later.

- o Invalid RBridge nickname (RBridge nickname is one of the reserved 0xFFC0 - 0xFFFF)
- o MTU mismatch
- o Invalid VLAN (Reserved VLANs)
- o 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 R Bridges. In such an environment, there may be large number of active paths between two end points. As an example, assume a topology with 4 R Bridges connected serially with 32 ECMP links at each hop. In the stated example topology, there are

$32 \times 32 \times 32 = 32768$  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 in to 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 R Bridge in the network. Detailed health report of a given region can be obtained from any R Bridge in the region.

An R Bridge associate itself with a region through its interfaces. A given interface can belong to one and only one region. An R Bridge can have multiple interfaces belonging to different regions. Each R Bridge is responsible for collecting monitoring data, organizing the data in to 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.

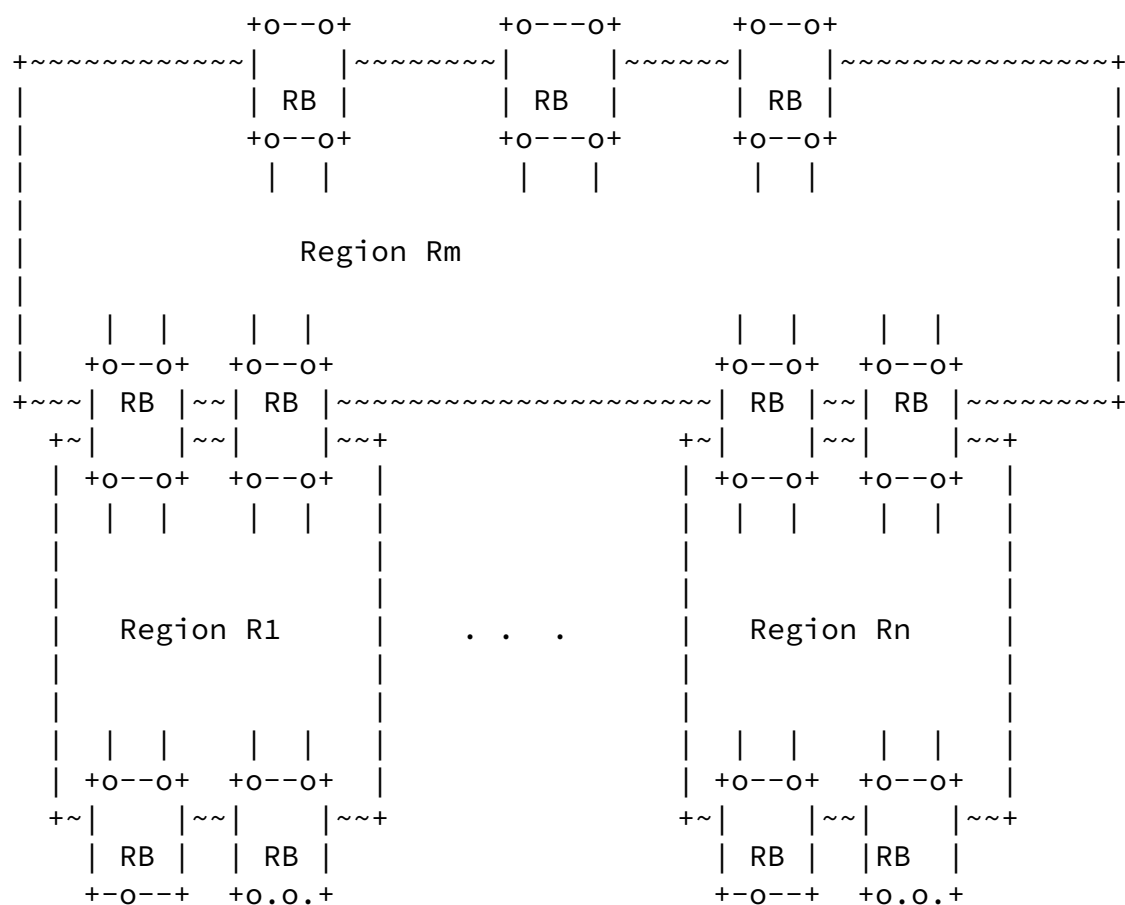


Figure 56 Example of "regions"

## 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 R Bridges 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.

Senevirathne

Expires January 6, 2013

[Page 87]

---

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

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:

- o On legacy interfaces
- o To neighbors not OAM capable
- o When ISIS state is not 2-way
- o When monitoring data advertisement is disabled

Rule 2: Distribution of Summary category data:

- o Distribute on all OAM capable interfaces
- o 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)



- o 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

- o Distributed on OAM capable interfaces
- o 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)
- o Do not advertise data element back in to the originator RBridge.

Each RBridge distribute data at periodic intervals. Each RBridge collects data it has received, analyzes them and redistribute according to the rules specified above. The distribution interval should be appropriately adjusted to not overload ISIS routing operations.

Then 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 RBridges 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 |                |
.---+                .
.                  .
|                  |
+-----+-----+
| #Rbrdridge|            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

Regiond-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.

- o Name of the region
- o Name of the RBridge
- o RBridge up time
- o Total number of neighbors
- o Total number of TRILL enabled ports in the RBridge
- o Total number of TRILL enabled ports Up
- o 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.

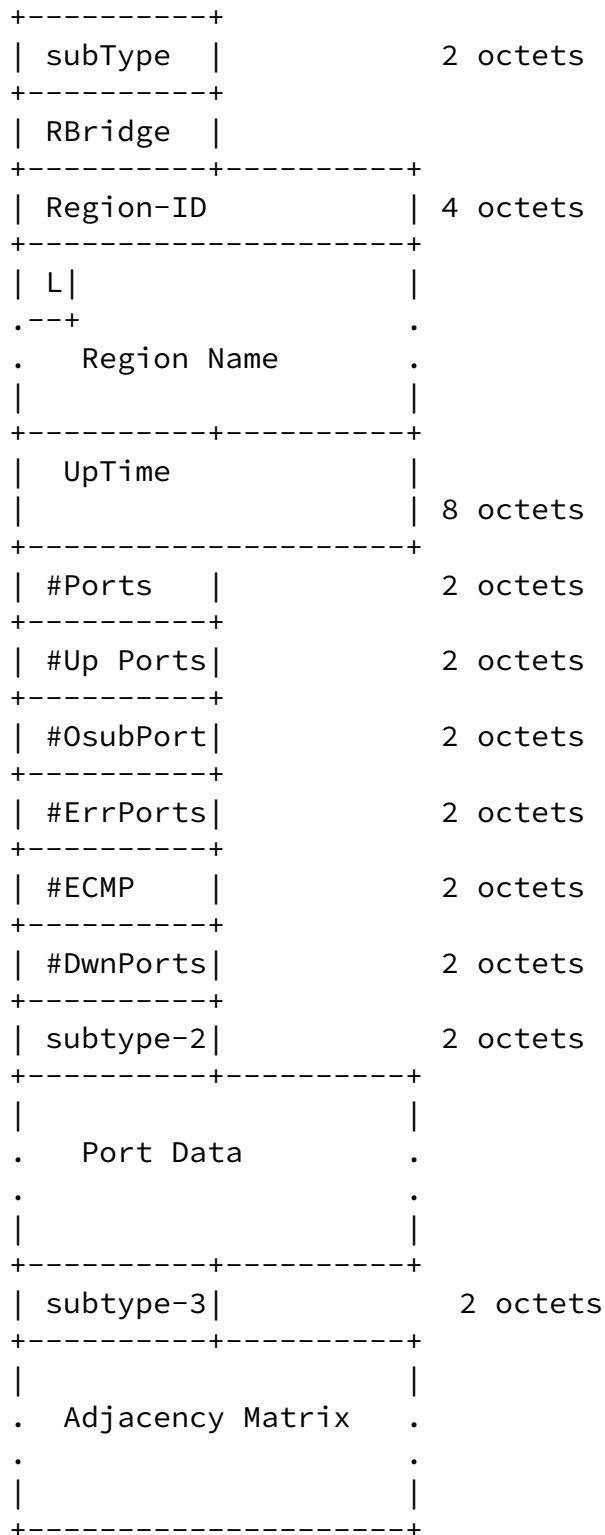


Figure 58 Encoding Detail Category Data

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

subtype-3: (2 octets): Set to 4. Following this sub type is the variable length Adjacency Matrix. See below for details

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

```

+-----+
| subType |                2 octets
+-----+
| RBridge |
+-----+-----+
| F |                1 octets
+-----+
| subtype-p|            2 octets
+-----+-----+
| ifindex          | 4 octets
+-----+-----+
| Slot      | Port      |
+-----+-----+
| Speed      | State      |
+-----+-----+

```

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.

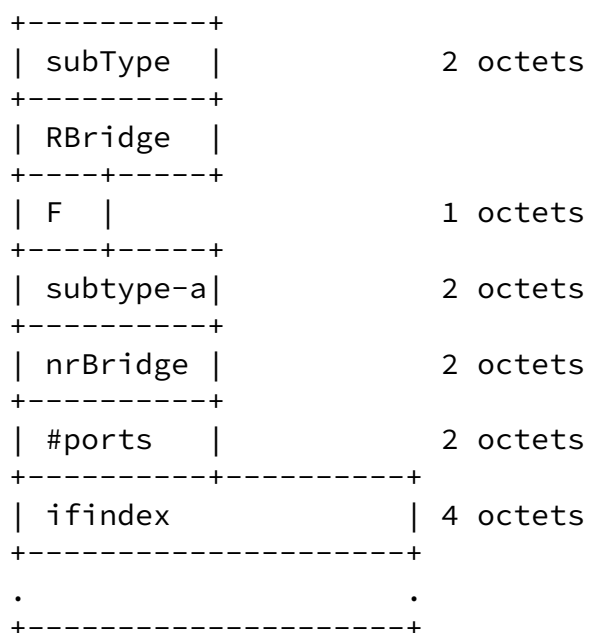


Figure 60 Encoding Adjacency Matrix

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.

```
+-----+
| subType |                2 octets
+-----+
| RBridge |                2 octets
+-----+-----+
| Region-ID |            4 octets
+---+-----+
| L |                    |
.---+                    .
.  Region Name            .
|                          |
+-----+-----+
| Vendor OUI |            4 octets
+-----+-----+
|              |
. Vendor specific          .
.  Information            .
|                          |
+-----+-----+
```

Figure 61 Encoding Vendor specific category Data

subType : (2 octets) set to 250 for Vendor specific category

RBridge: (2 octets) TRILL RBridge nickname [RFCtrill]

Regiond-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.

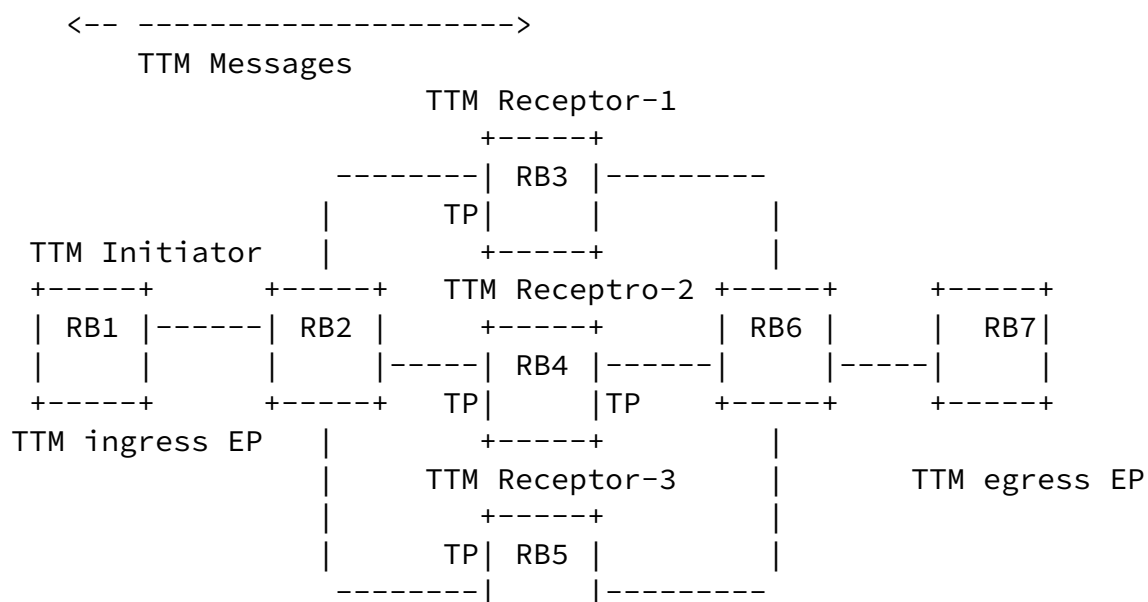


Figure 62 Traffic Triggered Monitoring

### [11.1. TTM Policy](#)

The TTM policy is a high level container that defines rules and actions. The TTM policy contains several sections.

- o TTM pattern
- o TTM mask
- o TTM Class
- o TTM frequency
- o TTM count
- o TTM actions
- o TTM Test Point
- o TTM Ingress Point
- o 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

- o count RX frames
- o count TX frames
- o count RX bytes
- o count TX bytes
- o count errors,
- o log,
- o 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:

- o TTM Set
- o TTM Get
- o TTM Remove
- o TTM Response
- o 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.



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

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC6325] Perlman, R. et.al. "Routing Bridges (RBridges): Base Protocol Specification", [RFC 6325](#), July 2011.
- [RFC6326] Eastlake, Donald. et.al. "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", [RFC 6326](#), July 2011.
- [RFC6327] Eastlake, Donald. et.al. "Routing Bridges (RBridges): Adjacency", [RFC 6327](#), July 2011.
- [RFC6165] Barnajee, A. and Ward, D." Extensions to IS-IS for Layer-2 Systems", [RFC 6165](#), April 2011.
- [GenApp] Ginsberg, L. et.al. "Advertising Generic Information in IS-IS", [draft-ietf-isis-genapp-04.txt](#), November, 2010.
- [RFC4884] Bonica, R. et.al "Extended ICMP to support Multi-Part messages", [RFC 4884](#), April, 2007.
- [RFC4379] Kompella, K, and Swallow, G. "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", [RFC 4379](#), February, 2006.

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

- [TRILLCH] Eastlake, Donald. et.al. "RBridges: TRILL RBridge Channel Support", [draft-ietf-trill-channel-02.txt](#), Work in Progress, July, 2011.
- [TRILLOAM] Bond, D. and Manral, V. "RBridges: Operations, Administration and Maintenance (OAM) Support", [draft-ietf-trill-rbridge-oam-00.txt](#), Work in Progress, July, 2011.
- [PINGEXT] Shen, N. et.al. "Traceroute and Ping Message Extensions", [draft-shen-traceroute-ping-msg-ext-03.txt](#), Work in Progress, October, 2011.
- [ISISMI] Previdi, S. et.al. "IS-IS Multi-Instance", [draft-ietf-isis-mi-05.txt](#), Work in Progress, October, 2011.

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

- [RFC792] Postel, J. "Internet Control Message Protocol (ICMP)", [RFC 792](#), September, 1981.
- [RFC826] Plummer, D. "Address Resolution Protocol", [RFC 826](#), November, 1982.
- [RFC2390] Bradley, T. et.al. "Inverse Address Resolution Protocol", September, [RFC 2390](#), 1988.
- [RFC5226] Narten, T. and Alverstand, H. "Guidelines for writing an IANA sections in RFCs", [RFC 5226](#), May 2008.

#### [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.

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

[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

| Region | Number<br>of switches | Max ECMP | Total#<br>Of Ports | % of Up<br>Ports | %of Ports<br>Oversubscribed | Err<br>Ports |
|--------|-----------------------|----------|--------------------|------------------|-----------------------------|--------------|
| xxx    | 40                    | 16       | 400                | 100              | 10                          | 1            |
| yyy    | 8                     | 2        | 25                 | 75               | 6                           | 0            |

### [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

| Switch | Next Hop switch | Interfaces                           |
|--------|-----------------|--------------------------------------|
| S1     | S2              | eth81,eth8/2(*),eth81<br>eth 10/2(x) |
|        | S3              | eth5/1 (?)                           |
|        | S4              | eth5/2,eth7/1                        |
| S2     | S1              | eth4/1,eth4/2,eth3/1<br>eth3/2(x)    |

#### [A.4.](#) C-Type usage in messages

The Table below lists various OAM messages and applicable mandatory and optional c-types.

| Message               | Mandatory Parameters                                             | OptionalParameters                                                                                                                                         |
|-----------------------|------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Loopback Request      | Version (1)                                                      | VLAN (4)<br>Service Tag (23)<br>Out-of-band<br>request Flag (1)<br>Reverse Path (26)<br>Control Plane<br>Verification (24)<br>Originator<br>IP address (2) |
| Loopback Response     | Version (1)<br>Cross Connect Error<br>Flag (1)<br>Final Flag (1) | Reverse Path<br>Response (27)<br>Control Plane<br>Response (25)                                                                                            |
| Path Trace<br>Request | Version (1)                                                      | VLAN (4)<br>Service Tag (23)<br>Out-of-band<br>request flag (1)                                                                                            |

|                     |                     |                   |
|---------------------|---------------------|-------------------|
|                     |                     | Reverse Path (26) |
|                     |                     | Control Plane     |
|                     |                     | Verification (24) |
|                     |                     | Originator        |
|                     |                     | IP address (2)    |
| +-----+-----+-----+ |                     |                   |
| Path Trace          | Version (1)         | Reverse Path      |
| Response            | Cross Connect Error | Response (27)     |
|                     | Flag (1)            | Control Plane     |
|                     | Final Flag (1)      | Response (25)     |
|                     | Upstream            |                   |
|                     | Identification (2)  |                   |
|                     | Downstream          |                   |
|                     | Identification (5)  |                   |
|                     | Path of this        |                   |
|                     | Payload (6)         |                   |
| +-----+-----+-----+ |                     |                   |

Figure 63 Optional and Mandatory c-types

| Message             | Mandatory Parameters | OptionalParameters |
|---------------------|----------------------|--------------------|
| +-----+-----+-----+ |                      |                    |
| Multicast Tree      | Version (1)          | VLAN (4)           |
| Verification        |                      | Service Tag (23)   |
| Request             |                      | In Scope (14)      |
|                     |                      | Control Plane      |
|                     |                      | Verification (24)  |
|                     |                      | Originator         |
|                     |                      | IP address (2)     |
| +-----+-----+-----+ |                      |                    |
| Multicast Tree      | Version (1)          | Control Plane      |
| Verification        | Cross Connect Error  | Response (25)      |
| Response            | Flag (1)             |                    |
|                     | Final Flag (1)       |                    |
|                     | Upstream             |                    |
|                     | Identification (2)   |                    |
|                     | Multicast Tree       |                    |

|         |                      |         |
|---------|----------------------|---------|
|         | Downstream List (15) |         |
|         | RBridge nickname(35) |         |
| +-----+ | +-----+              | +-----+ |

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](mailto:tsenevir@cisco.com)

Senevirathne

Expires January 6, 2013

[Page 109]

Internet-Draft

[draft-tissa-trill-oam-04.txt](#)

July 2012

Dinesh G Dutt  
CISCO Systems  
3800 Zankar Road  
San Jose, CA 95134

Email: [ddutt@cisco.com](mailto:ddutt@cisco.com)

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

Phone: 408-447-0000  
Email: [vishwas.manral@hp.com](mailto:vishwas.manral@hp.com)



Sam Aldrin  
Huawei Technologies  
2330 Central Express Way  
Santa Clara, CA 95051

Email: [aldrin.ietf@gmail.com](mailto:aldrin.ietf@gmail.com)