

TRILL Working Group  
Internet Draft  
Intended Status: Standard Track  
Expires April 2014

Deepak Kumar  
Samer Salam  
Tissa Senevirathne  
Cisco  
Oct 19, 2013

TRILL OAM MIB  
draft-deepak-trill-oam-mib-01.txt

#### Status of this Memo

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

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

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

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

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

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

#### Copyright Notice

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

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

#### Abstract

This defines Management Information Base (MIB) for the IETF TRILL (Transparent Interconnection of Lots of Links) OAM objects.



## Table of Contents

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

## 1. Introduction

The general framework for TRILL OAM is specified in [TRILLOAMFRM]. The details of the Fault Management [FM] solution, conforming to that framework, are presented in [TRILLOAMFM]. The solution leverages the message format defined in Ethernet Connectivity Fault Management (CFM) [802.1Q] as the basis for the TRILL OAM message channel.

This document uses the CFM MIB modules defined in [802.1Q] as the basis for TRILL OAM MIB, and augments the existing tables to add new TRILL managed objects. The document further defines a new table with associated managed objects for TRILL OAM specific functionality.

## 2. The Internet-Standard Management Framework

For a detailed overview of the Internet-Standard Management Framework, please refer to [RFC3410]. Managed objects are accessed

via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the Structure of Management Information (SMI) specification. This memo specifies a MIB module that is compliant to SMIV2 [RFC2578], [RFC2579] and [RFC2580].

### 3. Overview

The TRILL-OAM-MIB module is intended to provide an overall framework for managing TRILL OAM. It leverages the IEEE8021-CFM-MIB and IEEE8021-CFM-V2-MIB modules defined in [802.1Q], and augments the Mep and Mep Db entries. It also adds a new table for TRILL OAM specific messages.

### 4. Conventions

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

### 5. Structure of the MIB module

Objects in this MIB module are arranged into subtrees. Each subtree is organized as a set of related objects. The various subtrees are shown below, supplemented with the required elements of the IEEE8021-CFM-MIB module.

#### 5.1. Textual Conventions

Textual conventions are defined to represent object types relevant to the TRILL OAM MIB.

#### 5.2. TRILL-OAM-MIB relationship to IEEE8021-TC-MIB

In TRILL, traffic labeling can be done using either a 12-bit VLAN or a 24-bit fine grain label.

IEEE8021-TC-MIB defines IEEE8021ServiceSelectorType with two values:

- 1 representing a vlanId, and
- 2 representing a 24 bit isid.

We propose to use value 2 for TRILL's fine grain label. As such, TRILL-OAM-MIB will import IEEE8021ServiceSelectorType, IEEE8021ServiceSelectorValueOrNone, and IEEE8021ServiceSelectorValue from IEEE8021-TC-MIB.

### 5.3. TRILL OAM MIB Tree

#### TRILL-OAM-MIB

```
|--trillOamNotifications
    |--trillOamFaultAlarm
|--trillOamMibObjects
    |--trillOamMep
        |--trillOamMepTable
        |--trillOamMepFlowCfgTable
        |--trillOamPtrTable
        |--trillOamMtrTable
        |--trillOamMepDbTable
```

#### 5.3.1. Notifications

Notification (fault alarm) is sent to the management entity with the OID of the MEP that has detected the fault.

#### 5.3.2. TRILL OAM MIB Per MEP Objects

The TRILL OAM MIB Per MEP Objects are defined in the `trillOamMepTable`. The `trillOamMepTable` augments the `dotlagCfmMepEntry` (please see section 6.1) defined in IEEE8021-CFM-MIB. It includes objects that are locally defined for an individual MEP and its associated Flow.

##### 5.3.2.1. `trillOamMepTable` Objects

o `trillOamMepRName` - This object contains the Rbridge Nickname as defined in RFC 6325 section 3.7.

o `trillOamMepPtmTid` - indicates the next sequence number/transaction identifier to be sent in a Path Trace message. The sequence number may be zero because it wraps around.

o `trillOamMepNextttMtmTid` - indicates the next sequence number/transaction identifier to be sent in a Multi-destination message. The sequence number may be zero because it wraps

around.

- o trillOamMepMepPtrIn - indicates the total number of valid, in-order, Path Trace Replies received.
- o trillOamMepMepPtrInOutOfOrder - indicates the total number of valid, out-of-order, Path Trace Replies received.
- o trillOamMepMepPtrOut - indicates the total number of valid Path Trace Replies transmitted.
- o trillOamMepMtrIn - indicates the total number of valid, in-order, Multi-destination Replies received.
- o trillOamMepMtrInOutOfOrder - indicates the total number of valid, out-of-order, Multi-destination Replies received.
- o trillOamMepMtrOut - indicates the total number of valid Multi-destination Replies transmitted.
- o trillOamMepTxLbmDestRName - indicates the target destination Rbridge NickName as defined in [RFC6325] section 3.7.
- o trillOamMepTxLbmHC - indicates the hop count field to be transmitted.
- o trillOamMepTxLbmReplyModeOob - True indicates that the Reply Mode of the Loopback message is requested to be out-of-band, and that the "Out of band IP address" TLV is to be transmitted. False indicates that in-band reply is transmitted.
- o trillOamMepTransmitLbmReplyIp - indicates the IP address to be transmitted in the "Out of band IP Address TLV" in the Loopback message.
- o trillOamMepTxLbmFlowEntropy - indicates the 128 bytes Flow entropy to be transmitted, as defined in [TRILLOAMFM].
- o trillOamMepTxPtmDestRName - indicates the target Destination Rbridge Nickname to be transmitted, as defined in [RFC6325] section 3.7.
- o trillOamMepTxPtmHC - indicates the hop count field to be transmitted.
- o trillOamMepTxPtmReplyModeOob - True indicates that the Reply Mode of the Path Trace message is requested to be out-of-band, and that the "Out of band IP address TLV" is to be transmitted.

False indicates that in-band reply is transmitted.

o trillOamMepTransmitPtmReplyIP - indicates the IP address to be transmitted in the "Out of band IP Address TLV" in the Path Trace message.

o trillOamMepTranmitPtmFlowEntropy - indicates the 128 bytes Flow entropy to be transmitted, as defined in [TRILLOAMFM].

o trillOamMepTxPtmStatus - A Boolean flag set to True by the MEP Path Trace Initiator State Machine or a MIB manager to indicate that another Path trace message is being transmitted. Reset to false by the MEP Initiator State Machine.

o trillOamMepTxPtmResultOK - Indicates the result of the operation, True : The Path Trace Message(s) will be (or has been) sent, False: The Path Trace Message(s) will not be sent.

o trillOamMepTxPtmMessages - The number of Path Trace messages to be transmitted.

o trillOamMepTxPtmSeqNumber - Indicates the Path Trace Transaction Identifier of the first PTM (to be) sent. The value returned is undefined if trillOamMepTxPtmResultOK is false.

o trillOamMepTxMtmTree - Indicates the Multi-destination Tree identifier as defined in RFC6325.

o trillOamMepTxMtmHC - Indicates the hop count field to be transmitted.

o trillOamMepTxMtmReplyModeOob - True indicates that the Reply of the Multi-destination message is requested to be out-of-band, and that the "Out of band IP address TLV" is to be transmitted. False indicates that in-band reply is transmitted.

o trillOamMepTransmitMtmReplyIp - the IP address to be transmitted in the "Out of band IP address TLV" in the Multi-destination message.

o trillOamMepTxMtmFlowEntropy - 128 Byte Flow Entropy to be transmitted, as defined in [TRILL-FM].

o trillOamMepTxMtmStatus - A Boolean flag set to True by the MEP Multi-Destination Initiator State Machine or a MIB manager to indicate that another Multicast trace message is being transmitted. Reset to False by the MEP Initiator State Machine.

- o trillOamMepTxMtmResultOK - Indicates the result of the operation: -True The Multi-destination Message(s) will be (or has been) sent. -False The Multi-destination Message(s) will not be sent.
- o trillOamMepTxMtmMessages -The number of Multi-Destination Messages to be transmitted.
- o trillOamMepTxMtmSeqNumber - The Sequence Number of the first Multi-destination message (to be) sent. The value returned is undefined if trillOamMepTxMtmResultOK is false.
- o trillOamMepTxMtmScopeList - The Multi-destination Rbridge Scope list, 2 octets per Rbridge.

#### 5.3.2.2. trillOamMepFlowCfgTable Objects

Each row in this table represents a Flow Configuration Entry for the associated MEP. The table uses four indices. The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific Flow Configuration Entry on the selected MEP. Some write-able objects in this table are only applicable in certain cases (as described under each object below), and attempts to write values for them in other cases will be ignored.

- o trillOamMepFlowCfgIndex - an index to the TRILL OAM Mep flow configuration table which indicates the specific Flow for the MEP. The index is never reused for other flow sessions on the same MEP while this session is active. The index value keeps increasing until it wraps to 0. This value can also be used in Flow-identifier TLV.
- o trillOamMepFlowCfgFlowEntropy - This is 96 bytes of flow entropy as described in [TRILL-FM].
- o trillOamMepFlowCfgDestRname - The target Rbridge nickname field to be transmitted as defined in [RFC6325] section 3.7.
- o trillOamMepFlowCfgFlowHC - indicates the time to live field to be transmitted.
- o trillOamMepFlowCfgRowStatus - indicates the status of row. The write-able columns in a row cannot be changed if the row is active. All columns MUST have a valid value before a row can be activated.

#### 5.3.2.3. trillOamPtrTable Objects



Each row in the table represents a Path Trace Reply Entry for the defined MEP and Transaction. This table uses four indices. The first three indices identify the MEP and the fourth index specifies the Transaction Identifier, and this transaction identifier uniquely identifies the response for a MEP which can have multiple flow.

- o trillOamMepPtrTransactionId - indicates Transaction identifier/sequence number returned by a previous transmit path trace message command, indicating which PTM's response is going to be returned.

- o trillOamPtrHC - indicates hop count field value for a returned PTR.

- o trillOamMepPtrFlag - indicates FCOI field value for a returned PTR.

- o trillOamMepPtrErrorcode - indicates the Return code and Return sub-code value for a returned PTR.

- o trillOamMepPtrTerminalMep - indicates a Boolean value stating whether the forwarded PTM reached a MEP enclosing its MA, as returned in the Terminal MEP flag field.

- o trillOamMepPtrNextEgressIdentifier - An integer field holding the last Egress Identifier returned in the PTR Upstream Rbridge nickname TLV of the PTR. The Last Egress identifies the Upstream Nickname.

- o trillOamMepPtrIngress - The value returned in the Ingress Action field of the PTM. The value ingNoTlv(0) indicates that no Reply Ingress TLV was returned in the PTM.

- o trillOamMepPtrIngressMac - indicates the MAC address returned in the ingress MAC address field.

- o trillOamMepIngressPortIdSubtype - indicates ingress Port ID. The format of this object is determined by the value of the trillOamMepPtrIngressPortIdSubtype object.

- o trillOamMepIngressPortId - indicates the ingress port ID. The format of this object is determined by the value of the trillOamMepPtrIngressPortId object.

- o trillOamMepPtrEgressPortIdSubtype - indicates the value returned in the Egress Action field of the PTM. The value ingNoTlv(0) indicates that no Reply Egress TLV was returned in the PTM.

- o trillOamMepPtrEgressPortId - indicates the egress port ID. The format of this object is determined by the value of trillOamMepPtrEgressPortId object.
- o trillOamMepPtrChassisIdSubtype - This object specifies the format for the Chassis ID returned in the Sender ID TLV of the PTR, if any. This value is ignored if the trillOamMepPtrChassisId has a length of 0.
- o trillOamMepPtrChassisId - indicates the chassis ID returned in the Sender ID TLV of the PTR, if any. The format of this object is determined by the value of the trillOamMepPtrChassisIdSubtype object.
- o trillOamMepPtrOrganizationSpecificTlv - indicates all Organization specific TLVs returned in the PTR, if any. Includes all octets including and following the TLV length field of each TLV, concatenated together.
- o trillOamMepPtrNextHopNicknames - indicates Next hop Rbridge List TLV returned in the PTR, if any. Includes all octets including and following the TLV length concatenated together.

#### 5.3.2.4. trillOamMtrTable Objects

This table includes Multi-destination Reply managed objects. Each row in the table represents a Multi-destination Reply Entry for the defined MEP and Transaction. This table uses five indices: The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific Transaction Identifier on the selected MEP. The fifth index is the receive order of Multi-destination replies. Some write-able objects in this table are only applicable in certain cases (as described under each object below), and attempts to write a value for them in other cases will be ignored.

- o trillOamMepMtrTransactionId - indicates Transaction identifier/sequence number returned by a previous transmit Multi-destination message command, indicating which MTM's response is going to be returned.
- o trillOamMepMtrReceiveOrder - indicates an index to distinguish among multiple MTR with same MTR Transaction Identifier field value. trillOamMepMtrReceiveOrder are assigned sequentially from 1, in the order that the Multi-destination Tree Initiator received the MTRs.

- o trillOamMepMtrFlag - indicates FCOI field value for a returned MTR.
- o trillOamMepMtrErrorCode - indicates return code and return sub code value for a returned MTR.
- o trillOamMepMtrLastEgressIdentifier - indicates an integer field holding the Last Egress Identifier returned in the MTR Upstream Rbridge Nickname TLV of the MTR. The Last Egress Identifier identifies the Upstream Nickname.
- o trillOamMepMtrIngress - indicates the value returned in the Ingress Action Field of the MTR. The value ingNoTlv(0) indicates that no Reply Ingress TLV was returned in the MTM.
- o trillOamMepMtrIngressMac - indicates the MAC address returned in the ingress MAC address field.
- o trillOamMepMtrIngressPortIdSubtype - indicates the ingress Port ID. The format of this object is determined by the value of the trillOamMepMtrIngressPortIdSubtype object.
- o trillOamMepMtrIngressPortId - indicates the ingress Port Id. The format of this object is determined by the value of the trillOamMepMtrIngressPortId object.
- o trillOamMepMtrEgress - indicates the value returned in the Egress Action field of the MTR. The value ingNoTlv(0) indicates that no Reply Egress TLV was returned in the MTR.
- o trillOamMepMtrEgressMac - indicates the MAC address returned in the egress MAC address field.
- o trillOamMepMtrEgressPortIdSubtype - indicates the egress Port ID. The format of this object is determined by the value of the trillOamMepMtrEgressPortIdSubtype object.
- o trillOamMepMtrEgressPortId - indicates the egress port ID. The format of this object is determined by the value of the trillOamMepMtrEgressPortId object.
- o trillOamMepMtrChassisIdSubtype - indicates the format of the chassis ID returned in the Sender ID TLV of the MTR, if any. The value is ignored if the trillOamMepMtrChassisId has length of 0.
- o trillOamMepMtrChassisId - indicates the chassis ID returned in the Sender ID TLV of the MTR, if any. The format of this

object is determined by the value of the trillOamMepMtrChassisIdSubtype object.

- o trillOamMepMtrOrganizationSpecificTlv - indicates all Organization specific TLVs returned in the MTR, if any. Includes all octets including and following the TLV length field of each TLV, concatenated together.

- o trillOamMepMtrNextHopNicknames - indicates next hop Rbridge List TLV returned in the PTR, if any. Includes all octets including and following the TLV length field of each TLV, concatenated together.

- o trillOamMepMtrNextHopTotalReceivers - indicates value indicating that MTR response contains Multicast receiver availability TLV.

- o trillOamMepMtrReceiverCount - indicates the number of Multicast receivers available on responding Rbridge on the VLAN specified by the diagnostic VLAN.

#### 5.3.2.4. trillOamMepDbTable Objects

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

- o trillOamMepDbFlowIndex - This object identifies the Flow. If the Flow Identifier TLV is received then index received can also be used.

- o trillOamMepCfgFlowEntropy - indicates 96 bytes of Flow entropy.

- o trillOamMepDbFlowState - indicates the operational state of the remote MEP (flow based) IFF state machines.

- o trillOamMepDbRmepFailedOkTime - indicates the time (sysUpTime) at which the Remote Mep Flow State machine last entered either the RMEP\_FAILED or RMEP\_OK state.

- o trillOamMepDbRbridgeName - indicates Remote MEP Rbridge Nickname.

## 6. Relationship to other MIB module

The IEEE8021-CFM-MIB, IEEE801-CFM-V2-MIB and LLDP-MIB contain

objects relevant to TRILL OAM MIB. Management objects contained in these modules are not duplicated here, to reduce overlap to the extent possible.

#### 6.1. Relationship to IEEE8021-CFM-MIB

TRILL OAM MIB Imports the following management objects from IEEE8021-CFM-MIB:

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

trillOamMepTable Augments dotlagCfmMepEntry. Implementation of IEEE-CFM-MIB is required as we are Augmenting the IEEE-CFM-MIB Table. Objects/Tables which are not applicable to TRILL implementation has to be handled by TRILL implementation back end and appropriate values as described in IEEE-CFM-MIB has to be returned.

#### 6.2. MIB modules required for IMPORTS

The following MIB module IMPORTS objects from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], IEEE-8021-CFM-MIB, LLDP-MIB.

#### 7. Definition of the TRILL OAM MIB module

TRILL-OAM-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY,  
OBJECT-TYPE,  
NOTIFICATION-TYPE,  
Counter32,

```
Unsigned32,
Integer32
    FROM SNMPv2-SMI
RowStatus,
TruthValue,
TimeStamp,
MacAddress
    FROM SNMPv2-TC
OBJECT-GROUP,
NOTIFICATION-GROUP,
MODULE-COMPLIANCE
    FROM SNMPv2-CONF
dotlagCfmMdIndex,
dotlagCfmMaIndex,
dotlagCfmMepIdentifier,
dotlagCfmMepEntry,
dotlagCfmMepDbEntry,
DotlagCfmIngressActionFieldValue,
DotlagCfmEgressActionFieldValue,
DotlagCfmRemoteMepState
    FROM IEEE8021-CFM-MIB
LldpChassisId,
LldpChassisIdSubtype,
LldpPortId
    FROM LLDP-MIB;

trilloamMib MODULE-IDENTITY
    LAST-UPDATED      "201310191200Z"
    ORGANIZATION      "TBD"
    CONTACT-INFO
        "E-mail:   dekumar@cisco.com
        Postal:    510 McCarthy Blvd
                  Milpitas, CA 95035
                  U.S.A.
        Phone:      +1 408 853 9760"
    DESCRIPTION
        "This MIB module contains the management objects for the
        management of Trill Services Operations, Administration
        and Maintenance.
        Initial version. Published as RFC xxxx.
```

---

#### Reference Overview

A number of base documents have been used to create the Textual Conventions MIB. The following are the abbreviations for the baseline documents:

[CFM] refers to 'Connectivity Fault Management', IEEE 802.1ag-2007, December 2007

[Q.840.1] refers to 'ITU-T Requirements and analysis for NMS-EMS management interface of Ethernet over Transport and Metro Ethernet Network (EoT/MEN)', March 2007

[Y.1731] refers to ITU-T Y.1731 'OAM functions and mechanisms for Ethernet based networks', February 2011

-----

#### Abbreviations Used

Term	Definition
CCM	Continuity Check Message
CFM	Connectivity Fault Management
CoS	Class of Service
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ITU-T	International Telecommunication Union - Telecommunication Standardization Bureau
MAC	Media Access Control
MA	Maintenance Association (equivalent to a MEG)
MD	Maintenance Domain (equivalent to a OAM Domain in MEF 17)
MD Level	Maintenance Domain Level (equivalent to a MEG level)
ME	Maintenance Entity
MEG	Maintenance Entity Group (equivalent to a MA)
MEG Level	Maintenance Entity Group Level (equivalent to MD Level)
MEP	Maintenance Association End Point or MEG End Point
MIB	Management Information Base
MIP	Maintenance Domain Intermediate Point or MEG Intermediate Point
MP	Maintenance Point. One of either a MEP or a MIP
OAM	Operations, Administration, and Maintenance On-Demand
OAM actions	that are initiated via manual intervention for a limited time to carry out diagnostics. On-Demand OAM can result in singular or periodic OAM actions during the diagnostic time interval
PDU	Protocol Data Unit
RFC	Request for Comment
SNMP	Simple Network Management Protocol
SNMP Agent	An SNMP entity containing one or more command responder and/or notification originator applications (along with their associated SNMP engine). Typically implemented in an NE.
SNMP Manager	An SNMP entity containing one or more command generator and/or notification receiver applications (along with their associated SNMP engine). Typically implemented in an EMS or NMS.

```

        TLV             Type Length Value, a method of encoding Objects
        UTC             Coordinated Universal Time
        UNI             User-to-Network Interface
        VLAN            Virtual LAN"
    REVISION            "201310191200Z"
    DESCRIPTION
        "Initial version. Published as RFC xxxx."
    ::= { mib-2 xxx }

-- RFC Ed.: assigned by IANA, see section 9 for details
--
-- *****
-- Object definitions in the TRILL OAM MIB Module
-- *****

trillOamNotifications OBJECT IDENTIFIER
    ::= { trillOamMib 0 }

trillOamMibObjects OBJECT IDENTIFIER
    ::= { trillOamMib 1 }

trillOamMibConformance OBJECT IDENTIFIER
    ::= { trillOamMib 2 }

-- *****
-- Groups in the TRILL OAM MIB Module
-- *****

trillOamMep OBJECT IDENTIFIER
    ::= { trillOamMibObjects 1 }

-- *****
-- TRILL OAM MEP Configuration
-- *****

trillOamMepTable OBJECT-TYPE
    SYNTAX             SEQUENCE OF TrillOamMepEntry
    MAX-ACCESS          not-accessible
    STATUS              current
    DESCRIPTION
        "This table is an extension of the dotlagCfmMepTable and rows
        are automatically added or deleted from this table based upon
        row creation and destruction of the dotlagCfmMepTable.

        This table represents the local MEP TRILL OAM configuration
        table. The primary purpose of this table is provide local

```



```

        parameters for the TRILL OAM function found in [TRILL-FM] and
        instantiated at a MEP."
REFERENCE      "[TRILL-FM]"
::= { trillOamMep 1 }

trillOamMepEntry OBJECT-TYPE
    SYNTAX      TrillOamMepEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION  "The conceptual row of trillOamMepTable."
    AUGMENTS    { dotlagCfmMepEntry }
    ::= { trillOamMepTable 1 }

TrillOamMepEntry ::= SEQUENCE {
    trillOamMepRName          Unsigned32,
    trillOamMepNextPtmTid     Unsigned32,
    trillOamMepNextMtmTid     Unsigned32,
    trillOamMepPtrIn          Counter32,
    trillOamMepPtrInOutOfOrder Counter32,
    trillOamMepPtrOut         Counter32,
    trillOamMepMtrIn          Counter32,
    trillOamMepMtrInOutOfOrder Counter32,
    trillOamMepMtrOut         Counter32,
    trillOamMepTxLbmDestRName Unsigned32,
    trillOamMepTxLbmHC        Unsigned32,
    trillOamMepTxLbmReplyModeOob TruthValue,
    trillOamMepTransmitLbmReplyIp OCTET STRING,
    trillOamMepTxLbmFlowEntropy OCTET STRING,
    trillOamMepTxPtmDestRName Unsigned32,
    trillOamMepTxPtmHC        Unsigned32,
    trillOamMepTxPtmReplyModeOob TruthValue,
    trillOamMepTransmitPtmReplyIp OCTET STRING,
    trillOamMepTxPtmFlowEntropy OCTET STRING,
    trillOamMepTxPtmStatus    TruthValue,
    trillOamMepTxPtmResultOK  TruthValue,
    trillOamMepTxPtmMessages  Integer32,
    trillOamMepTxPtmSeqNumber Unsigned32,
    trillOamMepTxMtmTree      Unsigned32,
    trillOamMepTxMtmHC        Unsigned32,
    trillOamMepTxMtmReplyModeOob TruthValue,
    trillOamMepTransmitMtmReplyIp OCTET STRING,
    trillOamMepTxMtmFlowEntropy OCTET STRING,
    trillOamMepTxMtmStatus    TruthValue,
    trillOamMepTxMtmResultOK  TruthValue,
    trillOamMepTxMtmMessages  Integer32,
    trillOamMepTxMtmSeqNumber Unsigned32,
    trillOamMepTxMtmScopeList OCTET STRING

```

```
}

trillOamMepRName OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65471)
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object contains Rbridge NickName of TRILL Rbridge as
        defined in RFC 6325 section 3.7."
    REFERENCE   "TRILL-FM and RFC 6325 section 3.7"
    ::= { trillOamMepEntry 1 }

trillOamMepNextPtmTid OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Next sequence number/transaction identifier to be sent in a
        Path Trace message. This sequence number can be zero because it
        wraps around. Implementation should be unique to identify
        Transaction Id for a MEP with multiple flows."
    REFERENCE   "TRILL-FM 11.1.1.1"
    ::= { trillOamMepEntry 2 }

trillOamMepNextMtmTid OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Next sequence number/transaction identifier to be sent in a
        Multi-destination message. This sequence number can be zero
        because it wraps around. Implementation should be unique to
        identify Transaction Id for a MEP with multiple flows."
    REFERENCE   "TRILL-FM 12.2.1"
    ::= { trillOamMepEntry 3 }

trillOamMepPtrIn OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Total number of valid, in-order Path Trace Replies received."
    REFERENCE   "TRILL-FM section 11"
    ::= { trillOamMepEntry 4 }

trillOamMepPtrInOutOfOrder OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
```

```
STATUS          current
DESCRIPTION
    "Total number of valid, out-of-order Path Trace Replies received."
REFERENCE "TRILL-FM section 11"
::= { trillOamMepEntry 5 }

trillOamMepPtrOut OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Total number of valid, Path Trace Replies transmitted."
REFERENCE "TRILL-FM section 11"
::= { trillOamMepEntry 6 }

trillOamMepMtrIn OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Total number of valid, in-order Multi-destination Replies
    received."
REFERENCE "TRILL-FM section 12"
::= { trillOamMepEntry 7 }

trillOamMepMtrInOutOfOrder OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Total number of valid, out-of-order Multi-destination Replies
    received."
REFERENCE "TRILL-FM section 12"
::= { trillOamMepEntry 8 }

trillOamMepMtrOut OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Total number of valid, Multi-destination Replies
    transmitted."
REFERENCE "TRILL-FM section 12"
::= { trillOamMepEntry 9 }

trillOamMepTxLbmDestRName OBJECT-TYPE
SYNTAX          Unsigned32 (0..65471)
MAX-ACCESS      read-create
```

```
STATUS          current
DESCRIPTION
    "The Target Destination Rbridge NickName Field as
    defined in RFC 6325 section 3.7 to be transmitted."
REFERENCE "TRILL-FM and RFC6325 section 3.7"
 ::= { trillOamMepEntry 10 }

trillOamMepTxLbmHC OBJECT-TYPE
SYNTAX          Unsigned32(1..63)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "The Hop Count to be transmitted.
    "
REFERENCE "TRILL-FM section 3"
 ::= { trillOamMepEntry 11 }

trillOamMepTxLbmReplyModeOob OBJECT-TYPE
SYNTAX          TruthValue
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "True Indicates that Reply of Lbm is out of band and
    out of band IP Address TLV is to be transmitted.
    False indicates that In band reply is transmitted."
REFERENCE "TRILL-FM 10.1.2.1"
 ::= { trillOamMepEntry 12 }

trillOamMepTransmitLbmReplyIp OBJECT-TYPE
SYNTAX          OCTET STRING
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "IP address for out of band IP Address TLV is to be transmitted."
REFERENCE "TRILL-FM 10.1.2.1"
 ::= { trillOamMepEntry 13 }

trillOamMepTxLbmFlowEntropy OBJECT-TYPE
SYNTAX          OCTET STRING
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "128 Byte Flow Entropy as defined in TRILL-FM to be transmitted."
REFERENCE "TRILL-FM section 3"
 ::= { trillOamMepEntry 14 }

trillOamMepTxPtmDestRName OBJECT-TYPE
SYNTAX          Unsigned32 (0..65471)
```

```
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "The Target Destination Rbridge NickName Field
    as defined in RFC 6325 section 3.7 to be transmitted."
REFERENCE "TRILL-FM and RFC6325 section 3.7"
::= { trillOamMepEntry 15 }

trillOamMepTxPtmHC OBJECT-TYPE
SYNTAX          Unsigned32 (1..63)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "The Hop Count field to be transmitted.
    "
REFERENCE "TRILL-FM section 3"
::= { trillOamMepEntry 16 }

trillOamMepTxPtmReplyModeOob OBJECT-TYPE
SYNTAX          TruthValue
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "True Indicates that Reply of Ptm is out of band and
    out of band IP Address TLV is to be transmitted.
    False indicates that In band reply is transmitted."
REFERENCE "TRILL-FM section 11"
DEFVAL          { false }
::= { trillOamMepEntry 17 }

trillOamMepTransmitPtmReplyIp OBJECT-TYPE
SYNTAX          OCTET STRING
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "IP address for out of band IP Address TLV is to be transmitted."
REFERENCE "TRILL-FM section 11"
::= { trillOamMepEntry 18 }

trillOamMepTxPtmFlowEntropy OBJECT-TYPE
SYNTAX          OCTET STRING
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "128 Byte Flow Entropy as defined in TRILL-FM to be transmitted."
REFERENCE "TRILL-FM section 3"
::= { trillOamMepEntry 19 }
```

trillOamMepTxPtmStatus OBJECT-TYPE  
SYNTAX TruthValue  
MAX-ACCESS read-create  
STATUS current  
DESCRIPTION  
    "A Boolean flag set to true by the MEP Path Trace Initiator State Machine or an MIB manager to indicate that another Ptm is being transmitted.  
    Reset to false by the MEP Initiator State Machine."  
REFERENCE "TRILL-FM section 11"  
DEFVAL { false }  
::= { trillOamMepEntry 20 }

trillOamMepTxPtmResultOK OBJECT-TYPE  
SYNTAX TruthValue  
MAX-ACCESS read-create  
STATUS current  
DESCRIPTION  
    "Indicates the result of the operation:  
    - true The Path Trace Message(s) will be (or has been) sent.  
    - false The Path Trace Message(s) will not be sent."  
REFERENCE "TRILL-FM section 11"  
DEFVAL { true }  
::= { trillOamMepEntry 21 }

trillOamMepTxPtmMessages OBJECT-TYPE  
SYNTAX Integer32 (1..1024)  
MAX-ACCESS read-create  
STATUS current  
DESCRIPTION  
    "The number of Path Trace messages to be transmitted."  
REFERENCE "TRILL-FM section 11"  
::= { trillOamMepEntry 22 }

trillOamMepTxPtmSeqNumber OBJECT-TYPE  
SYNTAX Unsigned32  
MAX-ACCESS read-create  
STATUS current  
DESCRIPTION  
    "The Path Trace Transaction Identifier of the first PTM (to be) sent. The value returned is undefined if trillOamMepTxPtmResultOK is false."  
REFERENCE "TRILL-FM section 11"  
::= { trillOamMepEntry 23 }

trillOamMepTxMtmTree OBJECT-TYPE  
SYNTAX Unsigned32  
MAX-ACCESS read-create

```
STATUS          current
DESCRIPTION
    "The Multi-destination Tree is identifier for tree as defined in
    RFC6325."
 ::= { trillOamMepEntry 24 }

trillOamMepTxMtmHC OBJECT-TYPE
SYNTAX          Unsigned32(1..63)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "The Hop Count field to be transmitted.
    "
REFERENCE "TRILL-FM section 3, RFC 6325 section 3"
 ::= { trillOamMepEntry 25 }

trillOamMepTxMtmReplyModeOob OBJECT-TYPE
SYNTAX          TruthValue
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "True Indicates that Reply of Mtm is out of band and
    out of band IP Address TLV is to be transmitted.
    False indicates that In band reply is transmitted."
REFERENCE "TRILL-FM section 12"
 ::= { trillOamMepEntry 26 }

trillOamMepTransmitMtmReplyIp OBJECT-TYPE
SYNTAX          OCTET STRING
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "IP address for out of band IP Address TLV is to be transmitted."
REFERENCE "TRILL-FM section 12"
 ::= { trillOamMepEntry 27 }

trillOamMepTxMtmFlowEntropy OBJECT-TYPE
SYNTAX          OCTET STRING
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "128 Byte Flow Entropy as defined in TRILL-FM to be transmitted."
REFERENCE "TRILL-FM section 3"
 ::= { trillOamMepEntry 28 }

trillOamMepTxMtmStatus OBJECT-TYPE
SYNTAX          TruthValue
MAX-ACCESS      read-create
```

```

    STATUS          current
    DESCRIPTION
        "A Boolean flag set to true by the MEP Multi Destination Initiator Sta
te
        Machine or an MIB manager to indicate that another Mtm is being
        transmitted.
        Reset to false by the MEP Initiator State Machine."
    REFERENCE "TRILL-FM section 12"
    DEFVAL       { false }
    ::= { trillOamMepEntry 29 }

trillOamMepTxMtmResultOK OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "Indicates the result of the operation:
        - true   The Multi-destination Message(s) will be (or has been) sent.
        - false  The Multi-destination Message(s) will not be sent."
    REFERENCE "TRILL-FM section 12"
    DEFVAL      { true }
    ::= { trillOamMepEntry 30 }

trillOamMepTxMtmMessages OBJECT-TYPE
    SYNTAX      Integer32 (1..1024)
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "The number of Multi Destination messages to be transmitted."
    REFERENCE "TRILL-FM section 12"
    ::= { trillOamMepEntry 31 }

trillOamMepTxMtmSeqNumber OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "The Multi-destination Transaction Identifier of the first MTM (to be)
        sent. The value returned is undefined if
        trillOamMepTxMtmResultOK is false."
    REFERENCE "TRILL-FM section 12"
    ::= { trillOamMepEntry 32 }

trillOamMepTxMtmScopeList OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "The Multi-destination Rbridge Scope list, 2 OCTET per Rbridge."

```



REFERENCE "TRILL-FM section 12"  
 ::= { trillOamMepEntry 33 }

```
-- *****
-- TRILL OAM Tx Measurement Configuration Table
-- *****
```

trillOamMepFlowCfgTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF TrillOamMepFlowCfgEntry  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION  
 "This table includes configuration objects and operations for  
 the Trill OAM [TRILL-FM].  
  
 Each row in the table represents a Flow configuration Entry for  
 the defined MEP. This table uses four indices. The first  
 three indices are the indices of the Maintenance Domain,  
 MaNet, and MEP tables. The fourth index is the specific Flow  
 configuration Entry on the selected MEP.  
  
 Some writable objects in this table are only applicable in  
 certain cases (as described under each object), and attempts to  
 write values for them in other cases will be ignored."  
 REFERENCE "[TRILL-FM]"  
 ::= { trillOamMep 2 }

trillOamMepFlowCfgEntry OBJECT-TYPE  
 SYNTAX TrillOamMepFlowCfgEntry  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION  
 "The conceptual row of trillOamMepFlowCfgTable."  
 INDEX  
 {  
 dotlagCfmMdIndex,  
 dotlagCfmMaIndex,  
 dotlagCfmMepIdentifier,  
 trillOamMepFlowCfgIndex  
 }  
 ::= { trillOamMepFlowCfgTable 1 }

TrillOamMepFlowCfgEntry ::= SEQUENCE {  
 trillOamMepFlowCfgIndex Unsigned32,  
 trillOamMepFlowCfgFlowEntropy OCTET STRING,  
 trillOamMepFlowCfgDestRName Unsigned32,  
 trillOamMepFlowCfgFlowHC Unsigned32,  
 trillOamMepFlowCfgRowStatus RowStatus

```

    }

trillOamMepFlowCfgIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (1..65535)
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An index to the Trill OAM Mep Flow Configuration table which
        indicates the specific Flow for the MEP.

        The index is never reused for other flow sessions on the same
        MEP while this session is active. The index value keeps
        increasing until it wraps to 0.
        This value can also be used in Flow-identifier TLV [TRILL-FM]"
    REFERENCE "TRILL-FM"
    ::= { trillOamMepFlowCfgEntry 1 }

trillOamMepFlowCfgFlowEntropy OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "This is 128 byte of Flow Entropy as described in
        TRILL OAM [TRILL-FM]."
    REFERENCE "TRILL-FM section 3"
    ::= { trillOamMepFlowCfgEntry 2 }

trillOamMepFlowCfgDestRName OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65471)
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "The Target Destination Rbridge NickName Field as
        defined in RFC 6325 section 3.7 to be transmitted."
    REFERENCE "TRILL-FM section 3 and RFC 6325 section 3.7"
    ::= { trillOamMepFlowCfgEntry 3 }

trillOamMepFlowCfgFlowHC OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "The Time to Live field to be transmitted.
        to be transmitted."
    REFERENCE "TRILL-FM section 3 and RFC 6325 section 3.7"
    ::= { trillOamMepFlowCfgEntry 4 }

trillOamMepFlowCfgRowStatus OBJECT-TYPE

```

SYNTAX                RowStatus  
 MAX-ACCESS           read-create  
 STATUS                current  
 DESCRIPTION  
     "The status of the row."

The writable columns in a row cannot be changed if the row is active. All columns MUST have a valid value before a row can be activated."

::= { trillOamMepFlowCfgEntry 5 }

```
-- *****
-- TRILL OAM Path Trace Reply Table
-- *****
```

trillOamPtrTable OBJECT-TYPE

SYNTAX                SEQUENCE OF TrillOamPtrEntry  
 MAX-ACCESS           not-accessible  
 STATUS                current  
 DESCRIPTION

"This table includes Path Trace Reply objects and operations for the Trill OAM [TRILL-FM]."

Each row in the table represents a Path Trace Reply Entry for the defined MEP and Transaction. This table uses four indices. The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific Transaction Identifier on the selected MEP.

Some writable objects in this table are only applicable in certain cases (as described under each object), and attempts to write values for them in other cases will be ignored."

REFERENCE             "TRILL-FM"  
 ::= { trillOamMep 3 }

trillOamPtrEntry OBJECT-TYPE

SYNTAX                TrillOamPtrEntry  
 MAX-ACCESS           not-accessible  
 STATUS                current  
 DESCRIPTION

"The conceptual row of trillOamPtrTable."

INDEX                 {  
                       dotlagCfmMdIndex,  
                       dotlagCfmMaIndex,  
                       dotlagCfmMepIdentifier,  
                       trillOamMepPtrTransactionId  
                       }

```

 ::= { trillOamPtrTable 1 }

TrillOamPtrEntry ::= SEQUENCE {
    trillOamMepPtrTransactionId      Unsigned32,
    trillOamMepPtrHC                  Unsigned32,
    trillOamMepPtrFlag                Unsigned32,
    trillOamMepPtrErrorCode           Unsigned32,
    trillOamMepPtrTerminalMep         TruthValue,
    trillOamMepPtrLastEgressId        Unsigned32,
    trillOamMepPtrIngress              DotlagCfmIngressActionFieldValue,

    trillOamMepPtrIngressMac           MacAddress,
    trillOamMepPtrIngressPortIdSubtype LldpPortId,
    trillOamMepPtrIngressPortId        LldpPortId,
    trillOamMepPtrEgress                DotlagCfmEgressActionFieldValue,
    trillOamMepPtrEgressMac             MacAddress,
    trillOamMepPtrEgressPortIdSubtype   LldpPortId,
    trillOamMepPtrEgressPortId          LldpPortId,
    trillOamMepPtrChassisIdSubtype      LldpChassisIdSubtype,
    trillOamMepPtrChassisId             LldpChassisId,
    trillOamMepPtrOrganizationSpecificTlv OCTET STRING,
    trillOamMepPtrNextHopNicknames      OCTET STRING
}

trillOamMepPtrTransactionId OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4294967295)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Transaction identifier/sequence number returned by a previous
        transmit path trace message command, indicating which PTM's
        response is going to be returned."
    REFERENCE    "TRILL-FM section 11"
    ::= { trillOamPtrEntry 1 }

trillOamMepPtrHC OBJECT-TYPE
    SYNTAX      Unsigned32 (1..63)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Hop Count field value for a returned PTR."
    REFERENCE    "TRILL-FM"
    ::= { trillOamPtrEntry 2 }

trillOamMepPtrFlag OBJECT-TYPE
    SYNTAX      Unsigned32 (0..15)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION

```

"FCOI (TRILL OAM Message TLV) field value for a returned PTR."

REFERENCE "TRILL-FM, 9.4.2.1"  
 ::= { trillOamPtrEntry 3 }

trillOamMepPtrErrorCode OBJECT-TYPE  
SYNTAX Unsigned32 (0..65535)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"Return Code and Return Sub code value for a returned PTR."  
REFERENCE "TRILL-FM, 9.4.2.1"  
 ::= { trillOamPtrEntry 4 }

trillOamMepPtrTerminalMep OBJECT-TYPE  
SYNTAX TruthValue  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"A boolean value stating whether the forwarded PTM reached a MEP enclosing its MA, as returned in the Terminal MEP flag of the Flags field."  
REFERENCE "TRILL-FM"  
 ::= { trillOamPtrEntry 5 }

trillOamMepPtrLastEgressId OBJECT-TYPE  
SYNTAX Unsigned32 (0..65535)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"An Integer field holding the Last Egress Identifier returned in the PTR Upstream Rbridge nickname TLV of the PTR.  
The Last Egress Identifier identifies the Upstream Nickname"  
REFERENCE "TRILL-FM 9.4.3.4"  
 ::= { trillOamPtrEntry 6 }

trillOamMepPtrIngress OBJECT-TYPE  
SYNTAX DotlagCfmIngressActionFieldValue  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The value returned in the Ingress Action Field of the PTM.  
The value ingNoTlv(0) indicates that no Reply Ingress TLV was returned in the PTM."  
REFERENCE "TRILL-FM 9.4.1"  
 ::= { trillOamPtrEntry 7 }

trillOamMepPtrIngressMac OBJECT-TYPE

SYNTAX           MacAddress  
MAX-ACCESS       read-only  
STATUS           current  
DESCRIPTION  
    "MAC address returned in the ingress MAC address field."  
REFERENCE        "TRILL-FM 9.4.1"  
::= { trillOamPtrEntry 8 }

trillOamMepPtrIngressPortIdSubtype OBJECT-TYPE

SYNTAX           LldpPortId  
MAX-ACCESS       read-only  
STATUS           current  
DESCRIPTION  
    "Ingress Port ID. The format of this object is determined by  
    the value of the trillOamMepPtrIngressPortIdSubtype object."  
REFERENCE        "TRILL-FM 9.4.1"  
::= { trillOamPtrEntry 9 }

trillOamMepPtrIngressPortId OBJECT-TYPE

SYNTAX           LldpPortId  
MAX-ACCESS       read-only  
STATUS           current  
DESCRIPTION  
    "Ingress Port ID. The format of this object is determined by  
    the value of the trillOamMepPtrIngressPortId object."  
REFERENCE        "TRILL-FM 9.4.1"  
::= { trillOamPtrEntry 10 }

trillOamMepPtrEgress OBJECT-TYPE

SYNTAX           DotlagCfmEgressActionFieldValue  
MAX-ACCESS       read-only  
STATUS           current  
DESCRIPTION  
    "The value returned in the Egress Action Field of the PTM.  
    The value ingNoTlv(0) indicates that no Reply Egress TLV was  
    returned in the PTM."  
REFERENCE        "TRILL-FM 9.4.1"  
::= { trillOamPtrEntry 11 }

trillOamMepPtrEgressMac OBJECT-TYPE

SYNTAX           MacAddress  
MAX-ACCESS       read-only  
STATUS           current  
DESCRIPTION  
    "MAC address returned in the egress MAC address field."  
REFERENCE        "TRILL-FM 9.4.1"  
::= { trillOamPtrEntry 12 }

```
trillOamMepPtrEgressPortIdSubtype OBJECT-TYPE
    SYNTAX      LldpPortId
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Egress Port ID. The format of this object is determined by
         the value of the trillOamMepPtrEgressPortIdSubtype object."
    REFERENCE    "TRILL-FM 9.4.1"
    ::= { trillOamPtrEntry 13 }

trillOamMepPtrEgressPortId OBJECT-TYPE
    SYNTAX      LldpPortId
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Egress Port ID. The format of this object is determined by
         the value of the trillOamMepPtrEgressPortId object."
    REFERENCE    "TRILL-FM 9.4.1"
    ::= { trillOamPtrEntry 14 }

trillOamMepPtrChassisIdSubtype OBJECT-TYPE
    SYNTAX      LldpChassisIdSubtype
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object specifies the format of the Chassis ID returned
         in the Sender ID TLV of the PTR, if any. This value is
         meaningless if the trillOamMepPtrChassisId has a length of 0."
    REFERENCE    "TRILL-FM 9.4.1"
    ::= { trillOamPtrEntry 15 }

trillOamMepPtrChassisId OBJECT-TYPE
    SYNTAX      LldpChassisId
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The Chassis ID returned in the Sender ID TLV of the PTR, if
         any. The format of this object is determined by the
         value of the trillOamMepPtrChassisIdSubtype object."
    REFERENCE    "TRILL-FM 9.4.1"
    ::= { trillOamPtrEntry 16 }

trillOamMepPtrOrganizationSpecificTlv OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (0..0 | 4..1500))
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "All Organization specific TLVs returned in the PTR, if
```

any. Includes all octets including and following the TLV  
Length field of each TLV, concatenated together."

REFERENCE "TRILL-FM 9.4.1"

::= { trillOamPtrEntry 17 }

trillOamMepPtrNextHopNicknames OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..0 | 4..1500))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Next hop Rbridge List TLV returned in the PTR, if  
any. Includes all octets including and following the TLV  
Length field of each TLV, concatenated together."

REFERENCE "TRILL-FM 9.4.3.5"

::= { trillOamPtrEntry 18 }

-- \*\*\*\*\*

-- TRILL OAM Multi Destination Reply Table

-- \*\*\*\*\*

trillOamMtrTable OBJECT-TYPE

SYNTAX SEQUENCE OF TrillOamMtrEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table includes Multi-destination Reply objects and  
operations for the Trill OAM [TRILL-FM].

Each row in the table represents a Multi-destination Reply  
Entry for the defined MEP and Transaction.

This table uses five indices.

The first three indices are the indices of the Maintenance Domain,  
MaNet, and MEP tables. The fourth index is the specific  
Transaction Identifier on the selected MEP.

The fifth index is the receive order of Multi-destination  
replies.

Some writable objects in this table are only applicable in  
certain cases (as described under each object), and attempts to  
write values for them in other cases will be ignored."

REFERENCE "TRILL-FM"

::= { trillOamMep 4 }

trillOamMtrEntry OBJECT-TYPE

SYNTAX TrillOamMtrEntry

MAX-ACCESS not-accessible

STATUS current



## DESCRIPTION

"The conceptual row of trillOamMtrTable."

```

INDEX      {
            dotlagCfmMdIndex,
            dotlagCfmMaIndex,
            dotlagCfmMepIdentifier,
            trillOamMepPtrTransactionId,
            trillOamMepMtrReceiveOrder
          }
 ::= { trillOamMtrTable 1 }

```

```

TrillOamMtrEntry ::= SEQUENCE {
    trillOamMepMtrTransactionId      Unsigned32,
    trillOamMepMtrReceiveOrder       Unsigned32,
    trillOamMepMtrFlag               Unsigned32,
    trillOamMepMtrErrorCode          Unsigned32,
    trillOamMepMtrLastEgressId       Unsigned32,
    trillOamMepMtrIngress            DotlagCfmIngressActionFieldValue,

    trillOamMepMtrIngressMac         MacAddress,
    trillOamMepMtrIngressPortIdSubtype LldpPortId,
    trillOamMepMtrIngressPortId      LldpPortId,
    trillOamMepMtrEgress             DotlagCfmEgressActionFieldValue,
    trillOamMepMtrEgressMac          MacAddress,
    trillOamMepMtrEgressPortIdSubtype LldpPortId,
    trillOamMepMtrEgressPortId       LldpPortId,
    trillOamMepMtrChassisIdSubtype    LldpChassisIdSubtype,
    trillOamMepMtrChassisId          LldpChassisId,
    trillOamMepMtrOrganizationSpecificTlv OCTET STRING,
    trillOamMepMtrNextHopNicknames   OCTET STRING,
    trillOamMepMtrReceiverAvailability TruthValue,
    trillOamMepMtrReceiverCount       TruthValue
}

```

trillOamMepMtrTransactionId OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"Transaction identifier/sequence number returned by a previous transmit Multi-destination message command, indicating which MTM's response is going to be returned."

REFERENCE "TRILL-FM section 12"

```
 ::= { trillOamMtrEntry 1 }
```

trillOamMepMtrReceiveOrder OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An index to distinguish among multiple MTR with same MTR Transaction Identifier field value. trillOamMepMtrReceiveOrder are assigned sequentially from 1, in the order that the Multi-destination Tree Initiator received the MTRs."

REFERENCE "TRILL-FM"

::= { trillOamMtrEntry 2 }

## trillOamMepMtrFlag OBJECT-TYPE

SYNTAX Unsigned32 (0..15)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"FCOI (TRILL OAM Message TLV) field value for a returned MTR."

REFERENCE "TRILL-FM, 9.4.2.1"

::= { trillOamMtrEntry 3 }

## trillOamMepMtrErrorCode OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"Return Code and Return Sub code value for a returned MTR."

REFERENCE "TRILL-FM, 9.4.2.1"

::= { trillOamMtrEntry 4 }

## trillOamMepMtrLastEgressId OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"An Integer field holding the Last Egress Identifier returned in the MTR Upstream Rbridge Nickname TLV of the MTR."

The Last Egress Identifier identifies the Upstream Nickname."

REFERENCE "TRILL-FM 9.4.3.4"

::= { trillOamMtrEntry 5 }

## trillOamMepMtrIngress OBJECT-TYPE

SYNTAX DotlagCfmIngressActionFieldValue

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The value returned in the Ingress Action Field of the MTR."

The value ingNoTlv(0) indicates that no Reply Ingress TLV was returned in the MTM."

REFERENCE "TRILL-FM 12.2.3"

```
::= { trillOamMtrEntry 6 }

trillOamMepMtrIngressMac OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "MAC address returned in the ingress MAC address field."
    REFERENCE       "TRILL-FM 12.2.3"
    ::= { trillOamMtrEntry 7 }

trillOamMepMtrIngressPortIdSubtype OBJECT-TYPE
    SYNTAX          LldpPortId
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Ingress Port ID. The format of this object is determined by
         the value of the trillOamMepMtrIngressPortIdSubtype object."
    REFERENCE       "TRILL-FM 12.2.3"
    ::= { trillOamMtrEntry 8 }

trillOamMepMtrIngressPortId OBJECT-TYPE
    SYNTAX          LldpPortId
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Ingress Port ID. The format of this object is determined by
         the value of the trillOamMepMtrIngressPortId object."
    REFERENCE       "TRILL-FM 12.2.3"
    ::= { trillOamMtrEntry 9 }

trillOamMepMtrEgress OBJECT-TYPE
    SYNTAX          DotlagCfmEgressActionFieldValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value returned in the Egress Action Field of the MTR.
         The value ingNoTlv(0) indicates that no Reply Egress TLV was
         returned in the MTR."
    REFERENCE       "TRILL-FM 12.2.3"
    ::= { trillOamMtrEntry 10 }

trillOamMepMtrEgressMac OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "MAC address returned in the egress MAC address field."
```

```
REFERENCE          "TRILL-FM 12.2.3"
::= { trillOamMtrEntry 11 }

trillOamMepMtrEgressPortIdSubtype OBJECT-TYPE
SYNTAX              LldpPortId
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
    "Egress Port ID. The format of this object is determined by
    the value of the trillOamMepMtrEgressPortIdSubtype object."
REFERENCE          "TRILL-FM 12.2.3"
::= { trillOamMtrEntry 12 }

trillOamMepMtrEgressPortId OBJECT-TYPE
SYNTAX              LldpPortId
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
    "Egress Port ID. The format of this object is determined by
    the value of the trillOamMepMtrEgressPortId object."
REFERENCE          "TRILL-FM 12.2.3"
::= { trillOamMtrEntry 13 }

trillOamMepMtrChassisIdSubtype OBJECT-TYPE
SYNTAX              LldpChassisIdSubtype
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
    "This object specifies the format of the Chassis ID returned
    in the Sender ID TLV of the MTR, if any. This value is
    meaningless if the trillOamMepMtrChassisId has a length of 0."
REFERENCE          "TRILL-FM 12.2.3"
::= { trillOamMtrEntry 14 }

trillOamMepMtrChassisId OBJECT-TYPE
SYNTAX              LldpChassisId
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
    "The Chassis ID returned in the Sender ID TLV of the MTR, if
    any. The format of this object is determined by the
    value of the trillOamMepMtrChassisIdSubtype object."
REFERENCE          "TRILL-FM 12.2.3"
::= { trillOamMtrEntry 15 }

trillOamMepMtrOrganizationSpecificTlv OBJECT-TYPE
SYNTAX              OCTET STRING (SIZE (0..0 | 4..1500))
MAX-ACCESS          read-only
```

```

STATUS          current
DESCRIPTION
    "All Organization specific TLVs returned in the MTR, if
    any. Includes all octets including and following the TLV
    Length field of each TLV, concatenated together."
REFERENCE        "TRILL-FM 12.2.3"
::= { trillOamMtrEntry 16 }

trillOamMepMtrNextHopNicknames OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE (0..0 | 4..1500))
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Next hop Rbridge List TLV returned in the PTR, if
    any. Includes all octets including and following the TLV
    Length field of each TLV, concatenated together."
REFERENCE        "TRILL-FM 9.4.3.5"
::= { trillOamMtrEntry 17 }

trillOamMepMtrReceiverAvailability OBJECT-TYPE
SYNTAX          TruthValue
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "True value indicates that MTR response contained
    Multicast receiver availability TLV"
REFERENCE        "TRILL-FM 9.4.3.6"
::= { trillOamMtrEntry 18 }

trillOamMepMtrReceiverCount OBJECT-TYPE
SYNTAX          TruthValue
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Indicates the number of Multicast receivers available on
    responding RBridge on the VLAN specified by the
    diagnostic VLAN."
REFERENCE        "TRILL-FM 9.4.3.6"
::= { trillOamMtrEntry 19 }

-- *****
-- TRILL OAM MEP Database Table
-- *****

trillOamMepDbTable OBJECT-TYPE
SYNTAX          SEQUENCE OF TrillOamMepDbEntry
MAX-ACCESS      not-accessible
STATUS          current

```

## DESCRIPTION

"This table is an extension of the dotlagCfmMepDbTable and rows are automatically added or deleted from this table based upon row creation and destruction of the dotlagCfmMepDbTable.

"

## REFERENCE

"[TRILL-FM]"

::= { trillOamMep 5 }

## trillOamMepDbEntry OBJECT-TYPE

SYNTAX TrillOamMepDbEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"The conceptual row of trillOamMepDbTable."

AUGMENTS {  
dotlagCfmMepDbEntry  
}

::= { trillOamMepDbTable 1 }

## TrillOamMepDbEntry ::= SEQUENCE {

trillOamMepDbFlowIndex	Unsigned32,
trillOamMepDbFlowEntropy	OCTET STRING,
trillOamMepDbFlowState	DotlagCfmRemoteMepState,
trillOamMepDbFlowFailedOkTime	TimeStamp,
trillOamMepDbRbridgeName	Unsigned32,
trillOamMepDbLastGoodSeqNum	Counter32

}

## trillOamMepDbFlowIndex OBJECT-TYPE

SYNTAX Unsigned32 (1..65535)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object identifies the Flow. If Flow Identifier TLV is received than index received can also be used.

"

REFERENCE "TRILL-FM"

::= {trillOamMepDbEntry 1 }

## trillOamMepDbFlowEntropy OBJECT-TYPE

SYNTAX OCTET STRING

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"128 byte Flow Entropy.

"

REFERENCE "TRILL-FM section 3."

```

 ::= {trillOamMepDbEntry 2 }

trillOamMepDbFlowState OBJECT-TYPE
    SYNTAX      DotlagCfmRemoteMepState
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The operational state of the remote MEP (flow based)
        IFF State machines. State Machine is running now per
        flow."
    REFERENCE "TRILL-FM"
    ::= {trillOamMepDbEntry 3 }

trillOamMepDbFlowFailedOkTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The Time (sysUpTime) at which the Remote Mep Flow state
        machine last entered either the RMEP_FAILED or RMEP_OK
        state.
        "
    REFERENCE "TRILL-FM"
    ::= {trillOamMepDbEntry 4 }

trillOamMepDbRbridgeName OBJECT-TYPE
    SYNTAX      Unsigned32(0..65471)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Remote MEP Rbridge Nickname"
    REFERENCE "TRILL-FM RFC 6325 section 3"
    ::= {trillOamMepDbEntry 5 }

trillOamMepDbLastGoodSeqNum OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Last Sequence Number received."
    REFERENCE "TRILL-FM 13.1"
    ::= {trillOamMepDbEntry 6}

-- *****
**
-- TRILL OAM MIB NOTIFICATIONS (TRAPS)
-- This notification is sent to management entity whenever a MEP loses/restore
S
-- contact with its peer Flow Meps
-- *****
**

```

```

trillOamFaultAlarm NOTIFICATION-TYPE
  OBJECTS          { trillOamMepDbFlowState }
  STATUS           current
  DESCRIPTION
    "A MEP Flow has a persistent defect condition.
    A notification (fault alarm) is sent to the management
    entity with the OID of the Flow that has detected the fault.

    The management entity receiving the notification can identify
    the system from the network source address of the
    notification, and can identify the Flow reporting the defect
    by the indices in the OID of the
    trillOamMepFlowIndex, and trillOamFlowDefect
    variable in the notification:

        dotlagCfmMdIndex - Also the index of the MEP's
                           Maintenance Domain table entry
                           (dotlagCfmMdTable).
        dotlagCfmMaIndex - Also an index (with the MD table index)
                           of the MEP's Maintenance Association
                           network table entry
                           (dotlagCfmMaNetTable), and (with the MD
                           table index and component ID) of the
                           MEP's MA component table entry
                           (dotlagCfmMaCompTable).
        dotlagCfmMepIdentifier - MEP Identifier and final index
                                into the MEP table (dotlagCfmMepTable).
        trillOamMepFlowCfgIndex - Index identifies
                                indicates the specific Flow for the MEP"
  REFERENCE        "TRILL-FM"
  ::= { trillOamNotifications 1 }

-- *****
**
-- TRILL OAM MIB Module - Conformance Information
-- *****
**

trillOamMibCompliances OBJECT IDENTIFIER
  ::= { trillOamMibConformance 1 }

trillOamMibGroups OBJECT IDENTIFIER
  ::= { trillOamMibConformance 2 }

-- *****
-- TRILL OAM MIB Units of conformance
-- *****

trillOamMepMandatoryGroup OBJECT-GROUP

```



```

OBJECTS      {
    trillOamMepRName,
    trillOamMepNextPtmTid,
    trillOamMepNextMtmTid,
    trillOamMepPtrIn,
    trillOamMepPtrInOutOfOrder,
    trillOamMepPtrOut,
    trillOamMepMtrIn,
    trillOamMepMtrInOutOfOrder,
    trillOamMepMtrOut,
    trillOamMepTxLbmDestRName,
    trillOamMepTxLbmHC,
    trillOamMepTxLbmReplyModeOob,
    trillOamMepTransmitLbmReplyIp,
    trillOamMepTxLbmFlowEntropy,
    trillOamMepTxPtmDestRName,
    trillOamMepTxPtmHC,
    trillOamMepTxPtmReplyModeOob,
    trillOamMepTransmitPtmReplyIp,
    trillOamMepTxPtmFlowEntropy,
    trillOamMepTxPtmStatus,
    trillOamMepTxPtmResultOK,
    trillOamMepTxPtmMessages,
    trillOamMepTxPtmSeqNumber,
    trillOamMepTxMtmTree,
    trillOamMepTxMtmHC,
    trillOamMepTxMtmReplyModeOob,
    trillOamMepTransmitMtmReplyIp,
    trillOamMepTxMtmFlowEntropy,
    trillOamMepTxMtmStatus,
    trillOamMepTxMtmResultOK,
    trillOamMepTxMtmMessages,
    trillOamMepTxMtmSeqNumber,
    trillOamMepTxMtmScopeList
}
STATUS      current
DESCRIPTION
    "Mandatory objects for the TRILL OAM MEP group."
 ::= { trillOamMibGroups 1 }

trillOamMepFlowCfgTableGroup OBJECT-GROUP
    OBJECTS      {
        trillOamMepFlowCfgFlowEntropy,
        trillOamMepFlowCfgDestRName,
        trillOamMepFlowCfgFlowHC,
        trillOamMepFlowCfgRowStatus
    }
    STATUS      current

```

## DESCRIPTION

"Trill OAM MEP Flow Configuration objects group."

::= { trillOamMibGroups 2 }

trillOamPtrTableGroup OBJECT-GROUP

```
OBJECTS
{
    trillOamMepPtrHC,
    trillOamMepPtrFlag,
    trillOamMepPtrErrorCode,
    trillOamMepPtrTerminalMep,
    trillOamMepPtrLastEgressId,
    trillOamMepPtrIngress,
    trillOamMepPtrIngressMac,
    trillOamMepPtrIngressPortIdSubtype,
    trillOamMepPtrIngressPortId,
    trillOamMepPtrEgress,
    trillOamMepPtrEgressMac,
    trillOamMepPtrEgressPortIdSubtype,
    trillOamMepPtrEgressPortId,
    trillOamMepPtrChassisIdSubtype,
    trillOamMepPtrChassisId,
    trillOamMepPtrOrganizationSpecificTlv,
    trillOamMepPtrNextHopNicknames
}
```

STATUS current

## DESCRIPTION

"Trill OAM MEP PTR objects group."

::= { trillOamMibGroups 3 }

trillOamMtrTableGroup OBJECT-GROUP

```
OBJECTS
{
    trillOamMepMtrFlag,
    trillOamMepMtrErrorCode,
    trillOamMepMtrLastEgressId,
    trillOamMepMtrIngress,
    trillOamMepMtrIngressMac,
    trillOamMepMtrIngressPortIdSubtype,
    trillOamMepMtrIngressPortId,
    trillOamMepMtrEgress,
    trillOamMepMtrEgressMac,
    trillOamMepMtrEgressPortIdSubtype,
    trillOamMepMtrEgressPortId,
    trillOamMepMtrChassisIdSubtype,
    trillOamMepMtrChassisId,
    trillOamMepMtrOrganizationSpecificTlv,
    trillOamMepMtrNextHopNicknames,
    trillOamMepMtrReceiverAvailability,
    trillOamMepMtrReceiverCount
}
```

```

        }
    STATUS          current
    DESCRIPTION
        "Trill OAM MEP MTR objects group."
    ::= { trillOamMibGroups 4 }

trillOamMepDbGroup OBJECT-GROUP
    OBJECTS {
        trillOamMepDbFlowIndex,
        trillOamMepDbFlowEntropy,
        trillOamMepDbFlowState,
        trillOamMepDbFlowFailedOkTime,
        trillOamMepDbRbridgeName,
        trillOamMepDbLastGoodSeqNum
    }

    STATUS          current
    DESCRIPTION
        "Trill OAM MEP DB objects group."
    ::= { trillOamMibGroups 5 }

trillOamNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
        trillOamFaultAlarm
    }
    STATUS current
    DESCRIPTION
        "Objects for Notification Group"
    ::= { trillOamMibGroups 6 }

-- *****
-- TRILL OAM MIB Module Compliance statements
-- *****

trillOamMibCompliance MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION
        "The compliance statement for the TRILL OAM MIB."
    MODULE          -- this module
    MANDATORY-GROUPS {
        trillOamMepMandatoryGroup,
        trillOamMepFlowCfgTableGroup,
        trillOamPtrTableGroup,
        trillOamMtrTableGroup,
        trillOamMepDbGroup,
        trillOamNotificationGroup
    }
    ::= { trillOamMibCompliances 1 }

```

```
-- Compliance requirement for read-only implementation.

trillOamMibReadOnlyCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "Compliance requirement for implementation that only
    provide read-only support for TRILL-OAM-MIB.
    Such devices can be monitored but cannot be configured
    using this MIB module
    "
  MODULE -- this module
  MANDATORY-GROUPS {
    trillOamMepMandatoryGroup,
    trillOamMepFlowCfgTableGroup,
    trillOamPtrTableGroup,
    trillOamMtrTableGroup,
    trillOamMepDbGroup,
    trillOamNotificationGroup
  }
  -- trillOamMepTable

  OBJECT trillOamMepTxLbmDestRName
  MIN-ACCESS read-only
  DESCRIPTION
    "Write access is not required."

  OBJECT trillOamMepTxLbmHC
  MIN-ACCESS read-only
  DESCRIPTION
    "Write access is not required."

  OBJECT trillOamMepTxLbmReplyModeOob
  MIN-ACCESS read-only
  DESCRIPTION
    "Write access is not required."

  OBJECT trillOamMepTransmitLbmReplyIp
  MIN-ACCESS read-only
  DESCRIPTION
    "Write access is not required."

  OBJECT trillOamMepTxLbmFlowEntropy
  MIN-ACCESS read-only
  DESCRIPTION
    "Write access is not required."

  OBJECT trillOamMepTxPtmDestRName
  MIN-ACCESS read-only
```

## DESCRIPTION

"Write access is not required."

OBJECT trillOamMepTxPtmHC

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required."

OBJECT trillOamMepTxPtmReplyModeOob

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required."

OBJECT trillOamMepTransmitPtmReplyIp

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required."

OBJECT trillOamMepTxPtmFlowEntropy

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required."

OBJECT trillOamMepTxPtmStatus

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required."

OBJECT trillOamMepTxPtmResultOK

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required."

OBJECT trillOamMepTxPtmMessages

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required."

OBJECT trillOamMepTxPtmSeqNumber

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required."

OBJECT trillOamMepTxMtmTree

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required."

OBJECT trillOamMepTxMtmHC  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT trillOamMepTxMtmReplyModeOob  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT trillOamMepTransmitMtmReplyIp  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT trillOamMepTxMtmFlowEntropy  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT trillOamMepTxMtmStatus  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT trillOamMepTxMtmResultOK  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT trillOamMepTxMtmMessages  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT trillOamMepTxMtmSeqNumber  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT trillOamMepTxMtmScopeList  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

-- trillOamMepFlowCfgTable

```
OBJECT trillOamMepFlowCfgFlowEntropy
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required."
```

```
OBJECT trillOamMepFlowCfgDestRName
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required."
```

```
OBJECT trillOamMepFlowCfgFlowHC
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required."
```

```
OBJECT trillOamMepFlowCfgRowStatus
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required."
```

```
::= { trillOamMibCompliances 2 }
```

END

## 8. Security Considerations

This MIB relates to a system which will provide network connectivity and packet forwarding services. As such, improper manipulation of the objects represented by this MIB may result in denial of service to a large number of end-users.

There are number of management objects defined in this MIB module with a MAX-ACCESS clause of read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have negative effect on sensitivity/vulnerability are described below.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control event GET and/or NOTIFY access to these objects and possibly to event encrypt the values of these objects when sending them over the network via SNMP.

SNMP version prior to SNMPv3 did not include adequate security. Even

if the network itself is secure, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanism (for authentication and privacy).

Further, deployment of SNMP version prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

## 9. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER value recorded in the SMI Numbers registry:

Descriptor	OBJECT	IDENTIFIER	value
-----			
trillOamMIB	{	mib-2 xxx }	

Editor's Note (to be removed prior to publication): the IANA is requested to assign a value for "xxx" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.

## 10. Conclusions

## 11. References

### 11.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.



[RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

[RFC6325] Perlman, R., et.al., "Routing Bridges (R Bridges): Base Protocol Specification", RFC 6325, July 2011.

[RFCfgl] D. Eastlake, M. Zhang, P. Agarwal, R. Perlman, D. Dutt, "TRILL: Fine-Grained Labeling", draft-ietf-trill-fine-labeling, work in progress.

## 11.2. Informative References

[3] Faber, T., Touch, J. and W. Yue, "The TIME-WAIT state in TCP and Its Effect on Busy Servers", Proc. Infocom 1999 pp. 1573-1583.

[Fab1999]Faber, T., Touch, J. and W. Yue, "The TIME-WAIT state in TCP and Its Effect on Busy Servers", Proc. Infocom 1999 pp. 1573-1583.

[TRILLOAMREQ] Senevirathne, T., et.al., "Requirements for Operations, Administration and Maintenance (OAM) in TRILL", draft-ietf-trill-oam-req, Work in Progress, November, 2012.

[TRILLOAMFM] Salam, S., et.al., "TRILL OAM Framework", draft-ietf-trill-oam-framework, Work in Progress, November, 2012.

[TRILL-FM] Senevirathne, T., et.al., "TRILL Fault Management", draft-tissa-trill-oam-fm, Work in Progress, February, 2013.

## 12. Acknowledgments

Copyright (c) 2013 IETF Trust and the persons identified as authors of the code. All rights reserved. Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

Copyright (c) 2013 IETF Trust and the persons identified as authors of the code. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

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

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

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

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

#### Authors' Addresses

Deepak Kumar  
Cisco  
510 McCarthy Blvd,  
Milpitas, CA 95035, USA  
Phone : +1 408-853-9760  
Email: dekkumar@cisco.com

Samer Salam  
Cisco  
595 Burrard St. Suite 2123  
Vancouver, BC V7X 1J1, Canada  
Email: ssalam@cisco.com

Tissa Senevirathne  
Cisco  
375 East Tasman Drive  
San Jose, CA 95134, USA  
Email: tsenevir@cisco.com

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

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

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

Status of this Memo

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

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

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

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

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

Copyright Notice

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

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

## Internet-Draft Directory Assisted TRILL Encapsulation

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

### Abstract

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

### Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 0.

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

### Table of Contents

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

## Internet-Draft Directory Assisted TRILL Encapsulation

### 1. Introduction

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

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

### 2. Terminology

AF           Appointed Forwarder RBridge port

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

DA:        Destination Address

DC:        Data Center

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

FDB:       Filtering Database for Bridge or Layer 2 switch

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

SA:        Source Address

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

VM:        Virtual Machines

### 3. Directory Assistance to Non-RBridge

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

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

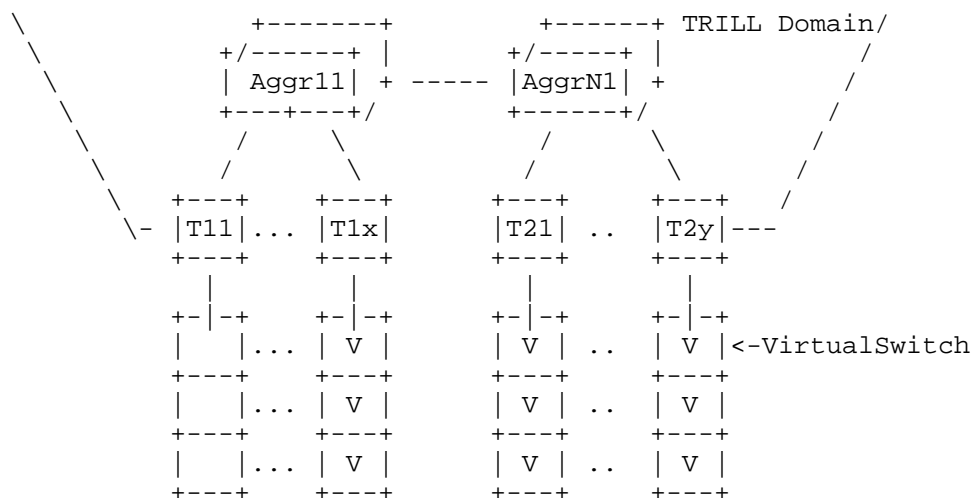


Figure 1: TRILL domain in typical Data Center Network

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

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

## Internet-Draft Directory Assisted TRILL Encapsulation

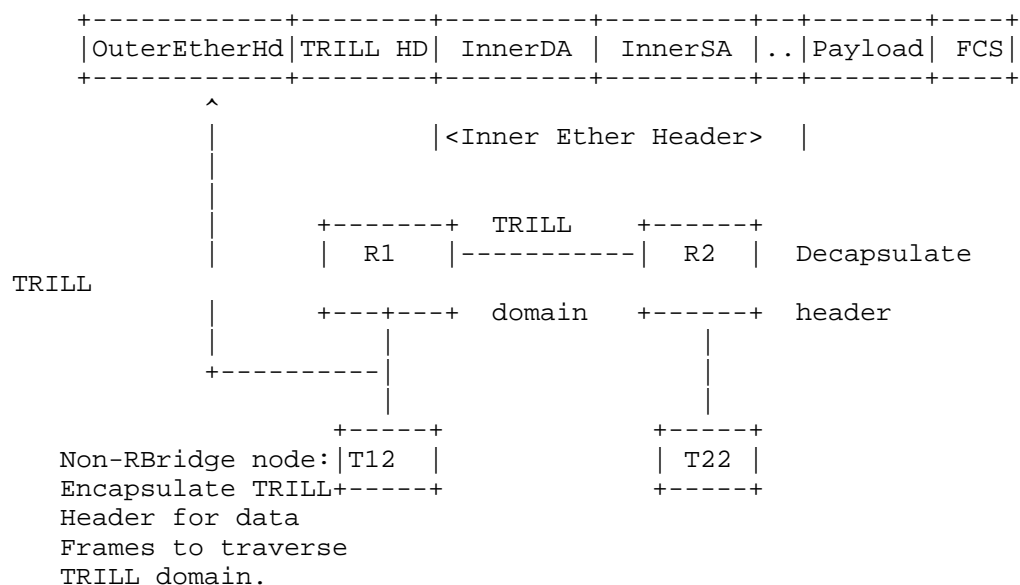
That means there is no need to worry about AF ports and all RBridge edge ports connected to one bridged LAN can receive and forward pre-encapsulated traffic, which greatly improves the overall network utilization.

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

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

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

We call a node that only performs the TRILL encapsulation but doesn't participate in RBridge's IS-IS routing a "TRILL Encapsulating node" or "Simplified RBridge". The TRILL Encapsulating Node gets the MAC&VLAN<->RBridgeEdge mapping table pushed down or pulled from directory servers [RBridge-directory]. Upon receiving a native Ethernet frame, the TRILL Encapsulating Node checks the MAC&VLAN<->RBridgeEdge mapping table, and perform the corresponding TRILL encapsulation if the entry is found in the mapping table. If the destination address and VLAN of the received Ethernet frame doesn't exist in the mapping table and no positive reply from pulling request to a directory, the Ethernet frame is forwarded per IEEE802.1Q.



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

The TRILL header includes a Source RBridge's Nickname (ingress) and Destination RBridge's Nickname (egress). When a TRILL header is added by a non-RBridge node, using the Ingress RBridge edge node's nickname in the source address field will make the ingress RBridge node receive TRILL frames with its own nickname in the frames' source address field, which can be confusing.

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

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



## 5. Benefits of Non-RBridge encapsulating TRILL header

### 5.1. Avoid Nickname Exhaustion Issue

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

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

### 5.2. Reduce FDB size for switches on Bridged LANs

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

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

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

## Internet-Draft Directory Assisted TRILL Encapsulation

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

### 6. Conclusion and Recommendation

When directory service is available, nodes outside TRILL domain become capable of encapsulating TRILL header for data frames destined for remote RBridges that is not on the same bridged LAN. The non-RBridge encapsulation approach is especially useful when there are a large number of servers in a data center equipped with hypervisor-based virtual switches. It is relatively easy for virtual switches, which are usually software based, to get directory assistance and perform network address encapsulation.

### 7. Manageability Considerations

TBD.

### 8. Security Considerations

TBD.

### 9. IANA Considerations

TBD

### 10. Acknowledgments

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

### 11. References

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

## Internet-Draft Directory Assisted TRILL Encapsulation

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

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

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

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

### Authors' Addresses

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

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

## Internet-Draft Directory Assisted TRILL Encapsulation

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

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

## Intellectual Property Statement

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

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

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

## Disclaimer of Liability

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

## Internet-Draft Directory Assisted TRILL Encapsulation

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

### Acknowledgment

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



INTERNET-DRAFT  
Intended status: Proposed Standard  
Updates: ESADI

Linda Dunbar  
Donald Eastlake  
Huawei  
Radia Perlman  
Intel  
Igor Gashinsky  
Yahoo  
Yizhou Li  
Huawei  
October 21, 2013

Expires: April 20, 2014

TRILL: Edge Directory Assistance Mechanisms  
<draft-dunbar-trill-scheme-for-directory-assist-06.txt>

#### Abstract

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

#### Status of This Memo

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

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

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

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

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

## Table of Contents

1. Introduction.....	3
1.1 Terminology.....	3
2. Push Model Directory Assistance Mechanisms.....	5
2.1 Requesting Push Service.....	5
2.2 Push Directory Servers.....	5
2.3 Push Directory Server State Machine.....	6
2.3.1 Push Directory States.....	6
2.3.2 Push Directory Events and Conditions.....	7
2.3.3 State Transition Diagram and Table.....	8
2.4 Additional Push Details.....	10
2.5 Primary to Secondary Server Push Service.....	11
3. Pull Model Directory Assistance Mechanisms.....	12
3.1 Pull Directory Request Format.....	12
3.2 Pull Directory Response Format.....	15
3.3 Pull Directory Hosted on an End Station.....	18
3.4 Pull Directory Request Errors.....	19
3.5 Cache Consistency.....	20
3.6 Additional Pull Details.....	22
4. Events That May Cause Directory Use.....	23
4.1 Forged Native Frame Ingress.....	23
4.2 Unknown Destination MAC.....	23
4.3 Address Resolution Protocol (ARP).....	24
4.4 IPv6 Neighbor Discovery (ND).....	25
4.5 Reverse Address Resolution Protocol (RARP).....	25
5. Layer 3 Address Learning.....	26
6. Directory Use Strategies and Push-Pull Hybrids.....	27
6.1 Strategy Configuration.....	27
7. Security Considerations.....	30
8. IANA Considerations.....	31
8.1 ESADI-Parameter Data.....	31
8.2 RBridge Channel Protocol Number.....	32
8.3 The Pull Directory and No Data Bits.....	32
Acknowledgments.....	33
Normative References.....	33
Informational References.....	34
Authors' Addresses.....	35



## 1. Introduction

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

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

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

The mechanisms used to initially populate directory data in primary servers is beyond the scope of this document. The Push Directory service can be used by a primary server to provide Directory data to secondary servers as described in Section 2.

### 1.1 Terminology

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

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

CP: Complete Push flag bit. See Sections 2 and 6.1 below.

CSNP Time: Complete Sequence Number PDU Time. See [ESADI] and Section 6.1 below.

Data Label: VLAN or FGL.

FGL: Fine Grained Label [RFCfgl].

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

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

PD: Push Directory flag bit. See Sections 2 and 6.1 below.

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

RBridge: An alternative name for a TRILL switch.

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

tenant: Sometimes used as a synonym for FGL.

TRILL switch: A device that implements the TRILL protocol.

## 2. Push Model Directory Assistance Mechanisms

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

With the Push Model, if complete address mapping information for a Data Label being pushed is available, a TRILL switch (RBridge) which has that complete pushed information can simply drop a native frame if the destination unicast MAC address can't be found in the mapping information available, instead of flooding it (see Section 2.1). This will minimize flooding of packets due to errors or inconsistencies but is not practical if directories have incomplete information.

### 2.1 Requesting Push Service

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

### 2.2 Push Directory Servers

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

For robustness, it is useful to have more than one copy of the data being pushed. Each RBridge that is a Push Directory server is configured with a number in the range 1 to 8, which defaults to 2, for each Data Label for which it can push directory information. If

the Push Directories for a Data Label are configured the same in this regard and enough such servers are available, this is the number of copies of the directory that will be pushed.

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

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

## 2.3 Push Directory Server State Machine

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

### 2.3.1 Push Directory States

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

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

Passive: No Push Directory data is advertised. Any outstanding EASDI-LSP fragments containing directory data are updated to remove that data and if the result is an empty fragment (contains nothing except possibly an Authentication TLV), the fragment is purged.

The Push Directory participates in [ESADI] and its [ESADI] fragment zero includes an ESADI-Parameters APPsub-TLV with the PD bit set to one and CP (Complete Push) bit zero.

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

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

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

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

### 2.3.2 Push Directory Events and Conditions

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

The Activate Condition: The server determines that there are K data reachable Push Directory servers, the server is configured that there should be N copies pushed, and K is less than or equal to N.

The Pacify Condition: The server determines that there are K data reachable Push Directory servers, the server is configured that there should be N copies pushed, and K is greater than N.

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

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

1. Push Directory server or TRILL switch was configured to be down but the TRILL switch on which it resides is up and the server is configured to be up.
2. The Push Directory server or the TRILL switch on which it is resident is being shut down.

3. The Activate Condition is met and the server is not configured to believe it has complete data.
4. The server determines that the Pacify Condition is met.
5. The server is configured to believe it has complete data and the Activate Condition is met.
6. The server is configured to believe it does not have complete data.
7. The Time Condition is met.

### 2.3.3 State Transition Diagram and Table

The state transition diagram is as follows.

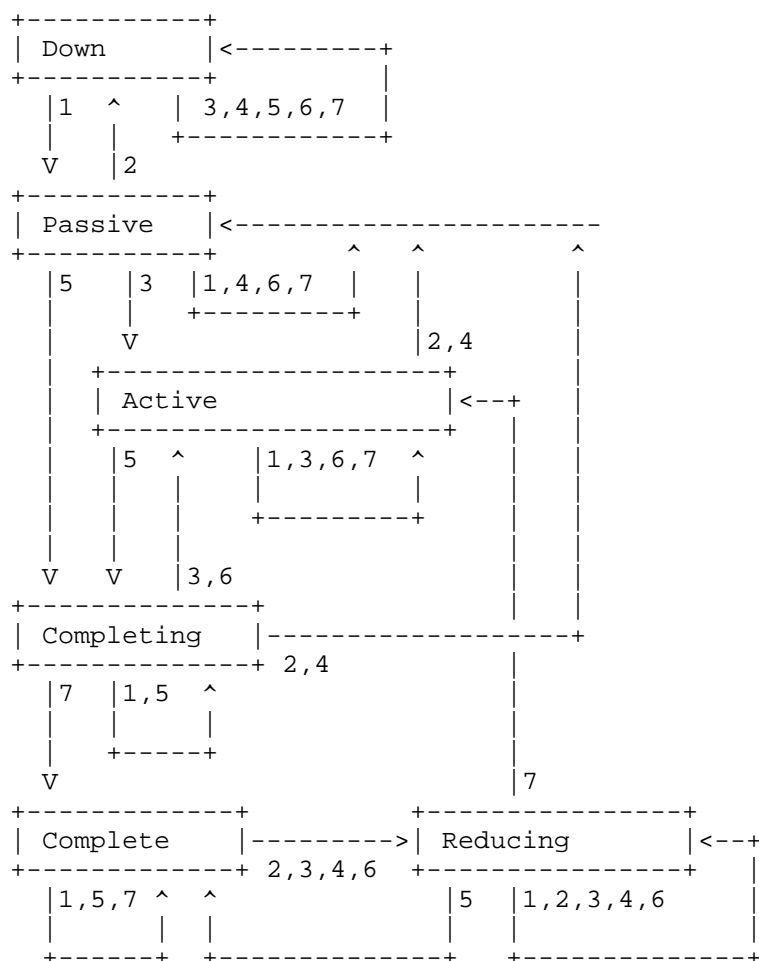


Figure 1. Push Server State Diagram

This state diagram is equivalent to the following transition table:

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

## 2.4 Additional Push Details

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

RBridges, whether or not they are a Push Directory server, MAY continue to advertise any locally learned MAC attachment information in [ESADI] using the Reachable MAC Addresses TLV [RFC6165] . However, if a Data Label is being served by complete Push Directory servers, advertising such locally learned MAC attachment should generally not be done as it would not add anything and would just waste bandwidth and ESADI link state space. An exception would be when an RBridge learns local MAC connectivity and that information appears to be missing from the directory mapping.

Because a Push Directory server may need to advertise interest in Data Labels even if it does not want to receive end station data in those Data Labels, the No Data flag bit is provided as discussed in Section 6.3.

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

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

1. The PD and CP bits are on in the ESADI zero fragment from the server for the relevant Data Label.
2. A client RBridge might be just coming up and receive an EASDI LSP meeting the requirement in point 1 above but have not yet received all of the ESADI LSP fragment from the Push Directory server. Thus, it should not believe that information to be complete unless it has also had data connectivity to the server for the larger of the client's and the server's CSNP times.

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



## 2.5 Primary to Secondary Server Push Service

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

### 3. Pull Model Directory Assistance Mechanisms

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

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

Pull Directory requests are sent by enclosing them in an RBridge Channel [Channel] message using the Pull Directory channel protocol number (see Section 8.2). Responses are returned in an RBridge Channel message using the same channel protocol number.

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

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

#### 3.1 Pull Directory Request Format

A Pull Directory request is sent as the Channel Protocol specific content of an inter-RBridge Channel message [Channel] TRILL Data packet. The Data Label in the packet is the Data Label in which the query is being made. The priority of the channel message is a mapping of the priority of the frame being ingressed that caused the request with the default mapping depending, per Data Label, on the strategy (see Section 6). The Channel Protocol specific data is formatted as follows:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   V   |   T   |   RESV   |   Count   |               RESV               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               Sequence Number               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| QUERY 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...
| QUERY 2
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...
| ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...
| QUERY K
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...

```

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

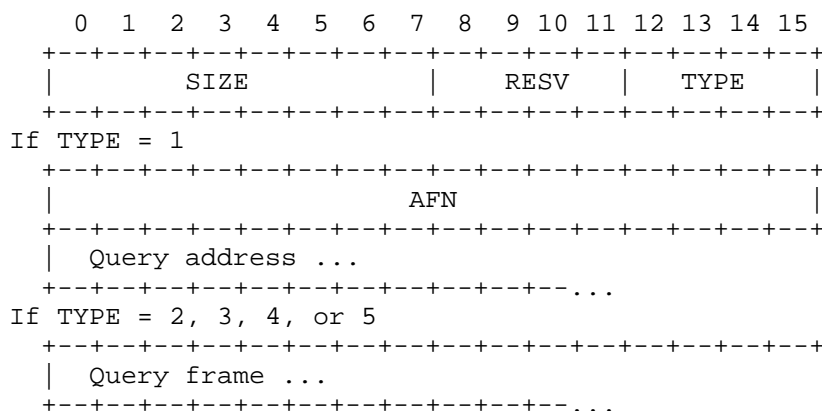
T: Type. 0 => Response, 1=> Query, 2=> Unsolicited Update, 3=> Reserved. An unsolicited update is formatted as a response except there is no corresponding query. Messages received with type = 3 are discarded. Queries received by an RBridge that is not a Pull Directory are discarded. Responses that do not match an outstanding Query are discarded.

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

Count: Number of queries present.

Sequence Number: A 32-bit quantity set by the sending RBridge, returned in any responses, and used to match up responses with queries. It is opaque except that the value zero is reserved for Unsolicited Update response messages. A Request received with Sequence Number zero is discarded.

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



SIZE: Size of the query data in bytes as an unsigned byte starting with and including the SIZE field itself. Thus the minimum legal value is 2. A value of SIZE less than 2 indicates a malformed message. The "QUERY" with the illegal SIZE value and all subsequent QUERYs MUST be ignored and the entire query message MAY be ignored.

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

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

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

AFN: Address Family Number of the query address.

Query Address: The query is asking for any other addresses, and the RBridge from which they are reachable, that correspond to the same interface, within the data label of the query. Typically that would be either (1) a MAC address with the querying RBridge primarily interested in the RBridge by which that MAC address is reachable, or

(2) an IP address with the querying RBridge interested in the corresponding MAC address and the RBridge by which that MAC address is reachable. But it could be some other address type.

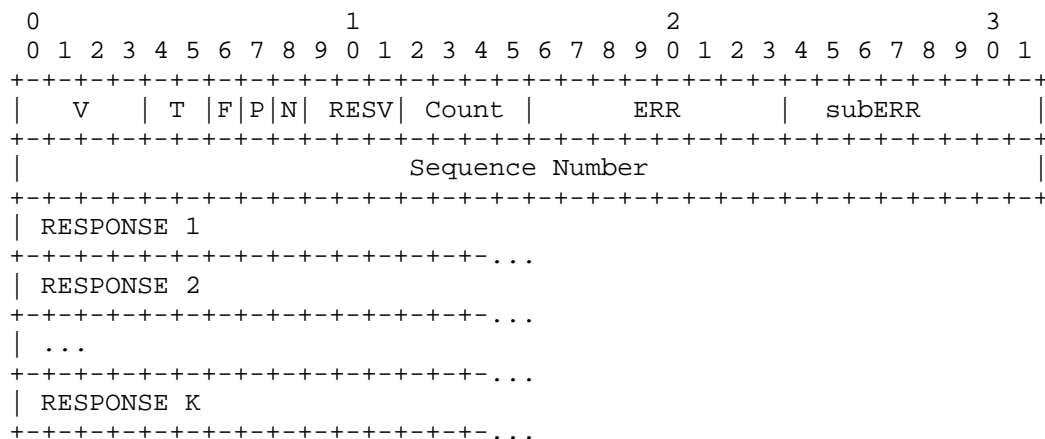
Query Frame: Where a Pull Directory query is the result of an ARP, ND, RARP, or unknown unicast MAC destination address, the ingress RBridge MAY send the frame to a Pull Directory Server if the frame is small enough to fit into a query message.

A query count of zero is explicitly allowed, for the purpose of pinging a Pull Directory server to see if it is responding to requests. On receipt of such an empty query message, a response message that also has a count of zero MUST be sent unless inhibited by rate limiting.

If no response is received to a Pull Directory request within a timeout configurable in milliseconds that defaults to 2,000, the request should be re-transmitted with the same Sequence Number up to a configurable number of times that defaults to three. If there are multiple queries in a request, responses can be received to various subsets of these queries by the timeout. In that case, the remaining unanswered queries should be re-sent in a new query with a new sequence number. If an RBridge is not capable of handling partial responses to requests with multiple queries, it MUST NOT send a request with more than one query in it.

### 3.2 Pull Directory Response Format

Pull Directory responses are sent as the Channel Protocol specific content of inter-RBridge Channel message TRILL Data packets. Responses are sent with the same Data Label and priority as the request to which they correspond except that the response priority is limited to be not more than a configured value. This priority limit is configurable at a per RBridge level and defaults to priority 6. The Channel protocol specific data format is as follows:



V, T: Version and Type as specified in Section 3.1.

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

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

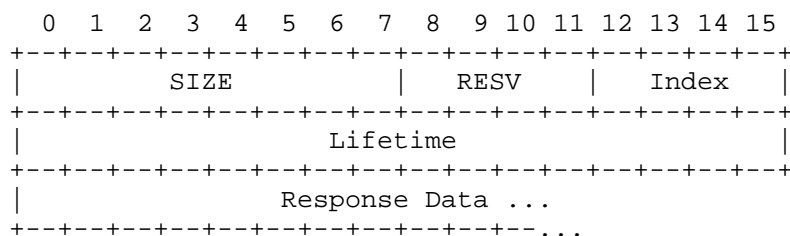
RESV: Reserved bits. MUST be sent as zero and ignored on receipt.

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

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

Sequence Number: A 32-bit quantity set by the sending RBridge, returned in any responses, and used to match up responses with queries. It is opaque except that the value zero is reserved for Unsolicited Update response messages.

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



SIZE: Size of the response data in bytes starting with and including the SIZE field itself. Thus the minimum value of SIZE is 6. If SIZE is less than 6, that RESPONSE and all subsequent RESPONSES MUST be ignored.

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

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

Lifetime: The length of time for which the response should be considered valid in seconds. If zero, the response can only be used for the particular query from which it resulted. The maximum time that can be expressed is just over 18.2 hours. [Perhaps this should be in units of, say, 200 milliseconds?]

Response Data: There are two types of response data.

If the ERR field is non-zero, the response data is a copy of the query data, that is, either an AFN followed by an address or a query frame.

If the ERR field is zero, the response data is the contents of an Interface Addresses APPsub-TLV (see Section 5) without the usual TRILL GENINFO TLV type and length and without the usual IA APPsub-TLV type and length before it. The maximum size of such contents is 251 bytes in the case when SIZE is 255.

Multiple response records can appear in a response message with the same index if the answer to a query consists of multiple Interface Address APPsub-TLV contents. This would be necessary if, for example, a MAC address within a Data Label appears to be reachable by multiple R Bridges. However, all RESPONSE records to any particular QUERY record MUST occur in the same response message. If a Pull Directory holds more mappings for a queried address than will fit into one response message, it selects which to include by some method outside the scope of this document.

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

### 3.3 Pull Directory Hosted on an End Station

Optionally, a Pull Directory actually hosted on an end station MAY be supported. In that case, when the RBridge advertising itself as a Pull Directory server receives a query, it modifies the inter-RBridge Channel message received into a native RBridge Channel message and forwards it to that end station. Later, when it receives one or more responses from that end station by native RBridge Channel messages, it modifies them into inter-RBridge Channel messages and forwards them to the source RBridge of the query.

The native Pull Directory RBridge Channel messages use the same Channel protocol number as do the inter-RBridge Pull Directory RBridge Channel messages. The native messages MUST be sent with an Outer.VLAN tag which gives the priority of each message which is the priority of the original inter-RBridge request packet. The Outer.VLAN ID used is the Designated VLAN on the link.

The native RBridge Channel message protocol dependent data for a Pull Directory query is formatted as follows:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  V   |  T   |  RESV   |  Count   |                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Data Label ... (4 or 8 bytes)                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Sequence Number                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  QUERY 1                                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  QUERY 2                                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  ...                                                             |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  QUERY K                                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

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

Nickname: The nickname of the RBridge sending the original inter-RBridge Pull Directory query.



All other fields, including the fields within the QUERY records are as specified in Section 3.1.

The native RBridge Channel message protocol specific content for a Pull Directory response is formatted as follows:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  V   |  T |F|P|N| RESV| Count |      ERR      |  subERR  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |
|      Nickname                      |
|                                     |
|  Data Label ... (4 or 8 bytes)    |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |
|      Sequence Number              |
+-----+-----+-----+-----+-----+-----+-----+-----+
| RESPONSE 1                        |
+-----+-----+-----+-----+-----+-----+-----+...
| RESPONSE 2                        |
+-----+-----+-----+-----+-----+-----+-----+...
| ...                              |
+-----+-----+-----+-----+-----+-----+-----+...
| RESPONSE K                        |
+-----+-----+-----+-----+-----+-----+-----+...

```

Nickname: If F=0, the nickname of the ultimate destination RBridge. If F=1, ignored.

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

All other fields, including the fields within the RESPONSE records, are as specified in Section 3.2.

### 3.4 Pull Directory Request Errors

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

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

If errors occur at the query level, they MUST be reported in a response message separate from the results of any successful queries.

If multiple queries in a request have different errors, they MUST be reported in separate response messages. If multiple queries in a request have the same error, this error response MAY be reported in one response message.

In an error response message, the query or queries being responded to appear, expanded by the Lifetime for which the server thinks the error might persist and with their Index inserted, as the RESPONSE record.

ERR values 1 through 127 are available for encoding request message level errors. ERR values 128 through 254 are available for encoding query level errors. the SubErr field is available for providing more detail on errors. The meaning of a SubErr field value depends on the value of the ERR field.

ERR	Meaning
---	-----
0	(no error)
1	Unknown or reserved field value
2	Request data too short
3-127	(Available for allocation by IETF Review)
128	Unknown AFN
129	Address not found
130-254	(Available for allocation by IETF Review)
255	Reserved

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

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

More TBD

### 3.5 Cache Consistency

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

A Pull Directory server MUST maintain one of the following, in order of increasing specificity. Retaining more specific records, such as that given in item 3 below, minimizes spontaneous response messages sent to update pull client RBridge caches. Retaining less specific records, such as that given in item 1, will generally increase the volume and overhead due to spontaneous response messages but still maintain consistency.

1. An overall record per Data Label of when the last positive response data will expire at a requester and when the last negative response will expire.
2. For each unit of data (IA APPsub-TLV Address Set [IA]) held by the server and each address about which a negative response was sent, when the last expected response with that data or negative response will expire at a requester.
3. For each unit of data held by the server and each address about which a negative response was sent, a list of RBridges that were sent that data as the response or sent a negative response for the address, with the expected time to expiration at each of them.

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

When data at a Pull Directory changes or is deleted or data is added and there may be unexpired stale information at a requesting RBridge, the Pull Directory MUST send an unsolicited message as discussed below.

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

discard of all positive and negative cached information for the Data Label.

If method 2 is being followed, then an RBridge floods address specific unsolicited update positive responses when data which is cached by a querying RBridge is changed or deleted and floods an address specific unsolicited update negative response when such information is added to. Such messages are similar to the method 1 flooded unsolicited flush messages. The U and F bits will be one and the message will be multicast. However that Count field will be non-zero and either the P or N bit, but not both, will be one. On receiving such as address specific message, if it is positive the addresses in the response records in the unsolicited response are compared to the addresses about which the recipient RBridge is holding cached positive or negative information and, if they match, the cached information is updated or the negative information replaced with the new positive information. On receiving an address specific unsolicited update negative response, the addresses in the response records in the unsolicited response are compared to the addresses about which the recipient RBridge is holding cached positive or negative information and, if they match, the any cached positive information is discarded.

If method 3 is being followed, the same sort of unsolicited update messages are sent as with method 2 except they are not normally flooded but unicast only to the specific RBridges the server believes may be holding the cached positive or negative information that may need updating. However, the Pull Directory server MAY flood the unsolicited update, for example if it determines that a sufficiently large fraction of its requesters need to be updated.

### 3.6 Additional Pull Details

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

Because a Pull Directory server may need to advertise interest in Data Labels even though it does not want to received end station data in those Data Labels, the No Data flag bit is provided as specified in Section 8.3.

#### 4. Events That May Cause Directory Use

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

Support for each of the uses below is separately optional.

##### 4.1 Forged Native Frame Ingress

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

##### 4.2 Unknown Destination MAC

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

### 4.3 Address Resolution Protocol (ARP)

Ingressing an ARP [RFC826]:

ARP is a flexible protocol. It is commonly used on a link to query for the MAC address corresponding to an IPv4 address, test if an IPv4 address is in use, or to announce a change in any of IPv4 address, MAC address, and/or point of attachment.

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

Examining the three types of ARP use:

#### 1. General ARP Request / Response

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

#### 2. Gratuitous ARP

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

#### 3. Address Probe ARP Query

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

#### 4.4 IPv6 Neighbor Discovery (ND)

Ingressing an IPv6 ND [RFC4861]:

TBD

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

#### 4.5 Reverse Address Resolution Protocol (RARP)

Ingressing a RARP [RFC903]:

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

## 5. Layer 3 Address Learning

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

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

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



## 6. Directory Use Strategies and Push-Pull Hybrids

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

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

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

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

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

### 6.1 Strategy Configuration

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

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

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

1. Use Push only. The RBridge subscribes to Push Directory data for Data Label X.
2. Use Pull only. When the RBridge ingresses a frame in Data Label X that can use Directory information, if it has cached information for the address it uses it. If it does not have either cached positive or negative information for the address, it sends a Pull Directory query.
3. Use Push and Pull. The RBridge subscribes to Push Directory data for Data Label X. When it ingresses a frame in Data Label X that can use Directory information and it does not find that information in its link state database of Push Directory information, it makes a Pull Directory query.

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

- A. Flood immediate. Flood the frame immediately (even if you are also sending a Pull Directory) request.
- B. Flood. Flood the frame immediately unless you are going to do a Pull Directory request, in which case you wait for the response or for the request to time out after retries and flood the frame if the request times out.
- C. Discard if complete or Flood immediate. If you have complete Push Directory information and the address is not in that information, discard the frame. If you do not have complete Push Directory information, the same as A above.
- D. Discard if complete or Flood. If you have complete Push Directory information and the address is not in that information, discard the frame. If you do not have complete Push Directory information, the same as B above.

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

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

Priority 7 is normally only used for urgent messages critical to adjacency and so is avoided by default for directory traffic.

## 7. Security Considerations

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

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

For general TRILL security considerations, see [RFC6325].

## 8. IANA Considerations

This section give IANA allocation and registry considerations.

### 8.1 ESADI-Parameter Data

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

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

Registration Procedures: Expert Review

References: [ESADI], This document

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

The CP bit is ignored if the PD bit is zero.

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

```

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

The meanings of all the fields are as specified in [ESADI] except that the added PushDirPriority is the priority of the advertising ESADI instance to be a Push Directory as described in Section 2.3. If

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

## 8.2 RBridge Channel Protocol Number

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

## 8.3 The Pull Directory and No Data Bits

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

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

An example of an RBridge serving as a directory that would not want multi-destination traffic in some Data Labels might be an RBridge that does not offer end station service for any of the Data Labels for which it is serving as a directory and is either (1) a Pull Directory or (2) a Push Directory for which all of the ESADI traffic can be handled by unicast [ESADI].

## Acknowledgments

The contributions of the following persons are gratefully acknowledged:

TBD

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

## Normative References

- [RFC826] - Plummer, D., "An Ethernet Address Resolution Protocol", RFC 826, November 1982.
- [RFC903] - Finlayson, R., Mann, T., Mogul, J., and M. Theimer, "A Reverse Address Resolution Protocol", STD 38, RFC 903, June 1984
- [RFC2119] - Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997
- [RFC3971] - Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC4861] - Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, October 2008.
- [RFC5310] - Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.
- [RFC6165] - Banerjee, A. and D. Ward, "Extensions to IS-IS for Layer-2 Systems", RFC 6165, April 2011.
- [RFC6325] - Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (Rbridges): Base Protocol Specification", RFC 6325, July 2011.
- [RFC5342bis] - Eastlake 3rd, D., "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters", BCP 141, RFC 5342, September 2008.
- [RFC6326bis] - Eastlake, D., Banerjee, A., Dutt, D., Perlman, R., and

A. Ghanwani, "TRILL Use of IS-IS", draft-ietf-isis-rfc6326bis, work in progress.

[RFCclear] - Eastlake, D., M. Zhang, A. Ghanwani, V. Manral, A. Banerjee, draft-ietf-trill-clear-correct-06.txt, in RFC Editor's queue.

[Channel] - D. Eastlake, V. Manral, Y. Li, S. Aldrin, D. Ward, "TRILL: RBridge Channel Support", draft-ietf-trill-rbridge-channel-08.txt, in RFC Editor's queue.

[RFCcgl] - D. Eastlake, M. Zhang, P. Agarwal, R. Perlman, D. Dutt, "TRILL: Fine-Grained Labeling", draft-ietf-trill-fine-labeling-07.txt, in RFC Editor's queue.

[ESADI] - Zhai, H., F. Hu, R. Perlman, D. Eastlake, O. Stokes, "TRILL (Transparent Interconnection of Lots of Links): The ESADI (End Station Address Distribution Information) Protocol", draft-ietf-trill-esadi, work in progress.

[IA] - Eastlake, D., L. Yizhou, R. Perlman, "TRILL: Interface Addresses APPsub-TLV", draft-eastlake-trill-ia-appsubtlv, work in progress.

#### Informational References

[RFC5227] - Cheshire, S., "IPv4 Address Conflict Detection", RFC 5227, July 2008.

[DirectoryFramework] - Dunbar, L., D. Eastlake, R. Perlman, I. Gashinsky, "TRILL Edge Directory Assistance Framework", draft-ietf-trill-directory-framework, in RFC Editor's queue.

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

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



Authors' Addresses

Linda Dunbar  
Huawei Technologies  
5430 Legacy Drive, Suite #175  
Plano, TX 75024, USA

Phone: (469) 277 5840  
Email: ldunbar@huawei.com

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

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

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

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

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

Email: igor@yahoo-inc.com

Yizhou Li  
Huawei Technologies  
101 Software Avenue,  
Nanjing 210012 China

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

## Copyright, Disclaimer, and Additional IPR Provisions

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

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



INTERNET-DRAFT  
Updates: RFCchannel  
Intended status: Proposed Standard  
Expires: April 20, 2014

Donald Eastlake  
Yizhou Li  
Huawei  
October 21, 2013

TRILL: RBridge Channel Tunnel Protocol  
<draft-eastlake-trill-channel-tunnel-00.txt>

## Abstract

The IETF TRILL (Transparent Interconnection of Lots of Links) protocol includes an optional mechanism, called RBridge Channel, for the transmission of typed messages between TRILL switches in the same campus and between TRILL switches and end stations on the same link. This document specifies optional extensions to RBridge Channel that provides three facilities: (1) A mechanism to send such messages between a TRILL switch and an end station in either direction, or between two end stations, when the two devices are in the same campus but not on the same link; (2) A method to support security facilities for RBridge Channel messages; and (3) A method to tunnel a variety of payload types by encapsulating them in an RBridge Channel message.

## Status of This Memo

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

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

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

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

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

## Table of Contents

1. Introduction.....	3
1.2. Terminology and Acronyms.....	3
2. Channel Tunnel Packet Format.....	5
3. Tunnel Payload Types.....	8
3.1 Null Payload.....	8
3.2 RBridge Channel Message Payload.....	8
3.3 TRILL Data Packet.....	9
3.4 TRILL IS-IS Packet.....	10
3.5 Ethernet Frame.....	11
3. Channel Tunnel Scopes.....	13
3.1 End Station to RBridge(s).....	14
4.2 RBridge to End Station.....	15
4.3 End Station to End Station.....	16
5. Security, Keying, and Algorithms.....	18
5.1 SType None.....	18
5.2 RFC 5310 Based Authentication.....	18
5.3 DTLS Based Security.....	19
6. Channel Tunnel Errors.....	20
6.1 SubERRs under ERR 6.....	20
7. IANA Considerations.....	21
8. Security Considerations.....	21
Normative References.....	22
Informative References.....	22
Acknowledgements.....	23
Authors' Addresses.....	24

## 1. Introduction

The IETF TRILL protocol [RFC6325] provides efficient least cost transparent frame routing in multi-hop networks with arbitrary topologies and link technologies, using link-state routing and a header with a hop count. End stations are attached to TRILL switches by Ethernet but links between TRILL switches can be arbitrary technology. In general, the TRILL way to address or specify a TRILL switch (RBridge) in a TRILL campus is by the switch's TRILL provided nickname [RFC6325] [ClearCorrect].

The TRILL protocol includes an optional RBridge Channel facility [RFCchannel] to support typed message transmission between two RBridges (for example BFD [RFCbfd]) in the same campus and between RBridges and end stations on the same link.

This document specifies optional extensions to RBridge Channel that provides three facilities:

- (1) A mechanism to send RBridge Channel messages between a TRILL switch and an end station in either direction, or between two end stations, when the two devices are in the same campus but not on the same link. This mechanism requires the cooperation of an RBridge that is on the same link as the end station or stations involved.
- (2) A method to support security facilities for RBridge Channel messages.
- (3) A method to tunnel a variety of payload types by encapsulating them in an RBridge Channel message.

Any one, two, or all three of these facilities can be use in the same message.

There is no mechanism to stop end stations on the same link, from sending native RBridge Channel messages to each other; however, such use is outside the scope of this document.

### 1.2. Terminology and Acronyms

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

This document uses the acronyms defined in [RFC6325] and [RFCchannel] supplemented by the following additional acronym:

Data Label - VLAN or Fine Grained Label [RFCfgl].

Primary Nickname - If a TRILL switch holds two or more nicknames, the one it holds with the highest priority is the primary nickname. If two or more are held with the same priority, the one with the lowest value, considered as a 16-bit unsigned integer in network byte order, is the primary nickname.

TRILL switch - An alternative term for an RBridge.

## 2. Channel Tunnel Packet Format

The general structure of an RBridge Channel message on a link between TRILL switches (RBridges) is shown in Figure 1 below. When a native RBridge Channel message is sent between an RBridge and an end station on the same link, in either direction, the TRILL Header (including the inner Ethernet addresses and Data Label) is omitted as shown in Figure 2. The type of RBridge Channel message is given by a Protocol field in the RBridge Channel Header which indicates how to interpret the Channel Protocol Specific Payload [RFCchannel].

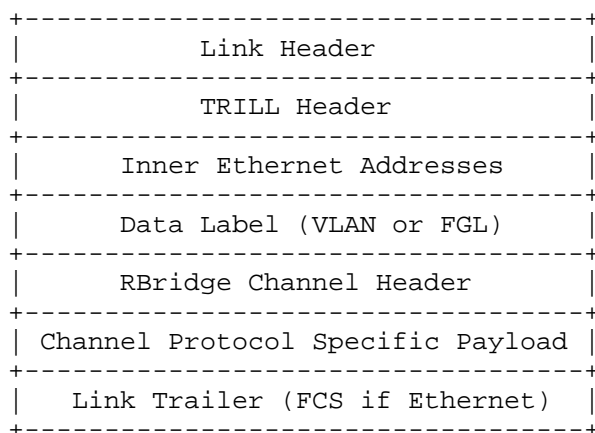


Figure 1. RBridge Channel Packet Structure

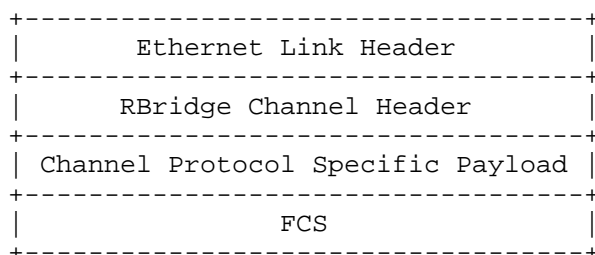


Figure 2. Native RBridge Channel Frame

The RBridge Channel Header looks like this:



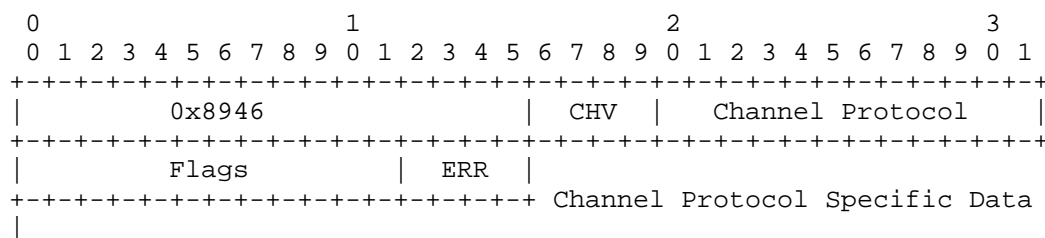


Figure 3. RBridge Channel Header

where 0x8946 is the RBridge Channel Ethertype and CHV is the Channel Header Version, currently zero.

The extensions specified herein are in the form of an RBridge Channel protocol, the Channel Tunnel Protocol. Figure 4 below expands the RBridge Channel Header and Protocol Specific Payload above for the case of the Channel Tunnel Protocol.

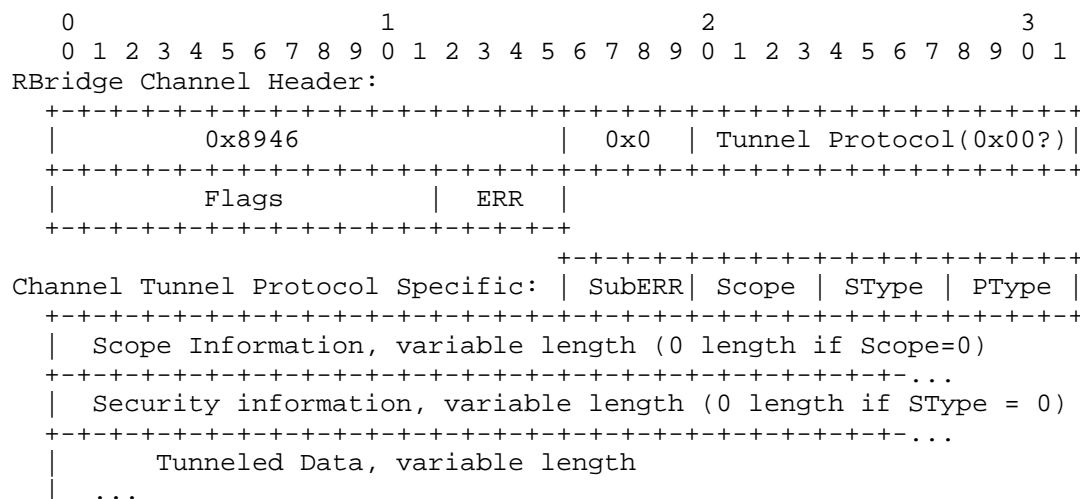


Figure 4. Channel Tunnel Header Structure

The RBridge Channel Header field specific to the RBridge Channel Tunnel Protocol is the Protocol field. Its contents MUST be the value allocated for this purpose (see Section 7).

The RBridge Tunnel Channel Protocol Specific fields are as follows:

**SubERR:** This field provides further details when a Tunnel Channel error is indicated in the RBridge Channel ERR field. If ERR is zero, then SubERR MUST be sent as zero and ignored on receipt. See Section 6.

Scope: This field describes the transport scope of the instance of Channel Tunnel. See Section 4.

SType: This field describes the type of security information and features, including keying material, being provided. See Section 5.

PType: Payload type. The describes the tunneled data. See Section 3 below.

The Channel Tunnel protocol is integrated with the RBridge Channel facility. Channel Tunnel errors are reported as if they were RBridge Channel errors, using newly allocated code points in the ERR field of the RBridge Channel Header supplemented by the SubErr field. Additional RBridge Channel Header flags are specified and used by Channel Tunnel.

### 3. Tunnel Payload Types

The RBridge Channel Tunnel Protocol can carry a variety of payloads as indicated by the PType field. Value are shown in the table below with further explanation after the table.

PType	Section	Description
0		Reserved
1	3.1	Null
2	3.2	RBridge Channel message
3	3.3	TRILL Data packet
4	3.4	TRILL IS-IS packet
5	3.5	Ethernet Frame
6-14		(Available for assignment by IETF Review)
15		Reserved

Table 1. Payload Type Values

While implementation of the Channel Tunnel protocol is optional, if it is implemented PTypes 1 (Null) and 2 (RBridge Channel message) MUST be implemented. PTypes 3, 4, and 5 MAY be implemented. The processing of any particular Channel Protocol message and its payload depends on meeting local security and other policy at the destination TRILL switch or end station.

#### 3.1 Null Payload

The Null payload type is intended to be used for messages such as key negotiation or the like. It indicates that there is no payload. Any data after the possible Scope Information and Security Information fields is ignored.

#### 3.2 RBridge Channel Message Payload

A PType of 2 indicates that the payload of the Channel Tunnel message is an encapsulated RBridge Channel message without the initial RBridge Channel Ethertype. Typical reasons for sending an RBridge Channel message inside a Channel Tunnel message are to provide security services, such as authentication or encryption, or to forward it through a cooperating border TRILL switch in either direction between an end station and a TRILL switch not on the same link.

This looks like the following:

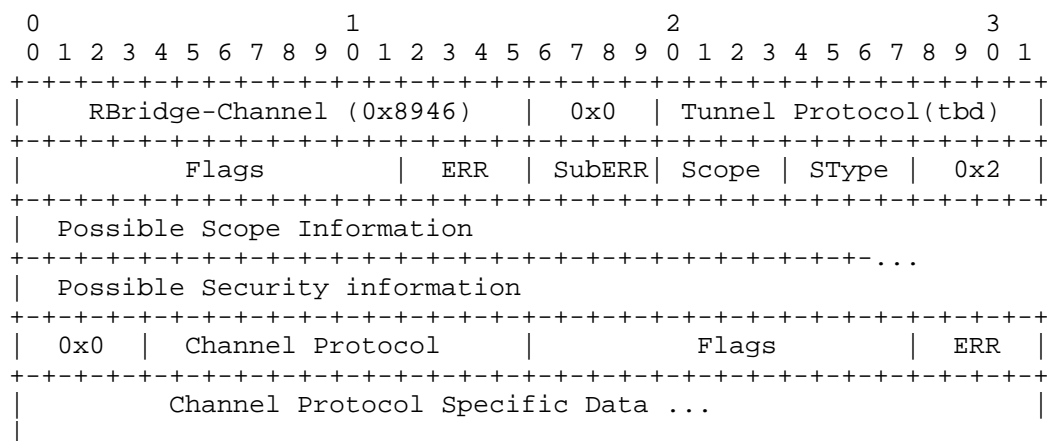


Figure 5. Tunneled Channel Message Channel Tunnel Structure

### 3.3 TRILL Data Packet

A PType of 3 indicates that the payload of the Tunnel protocol message is an encapsulated TRILL Data packet without the initial TRILL Ethertype as shown in the figure below. If this PType is implemented, the tunneled TRILL Data packet is handled as if it had been received by the destination TRILL switch on the port where the Channel Tunnel message was received.

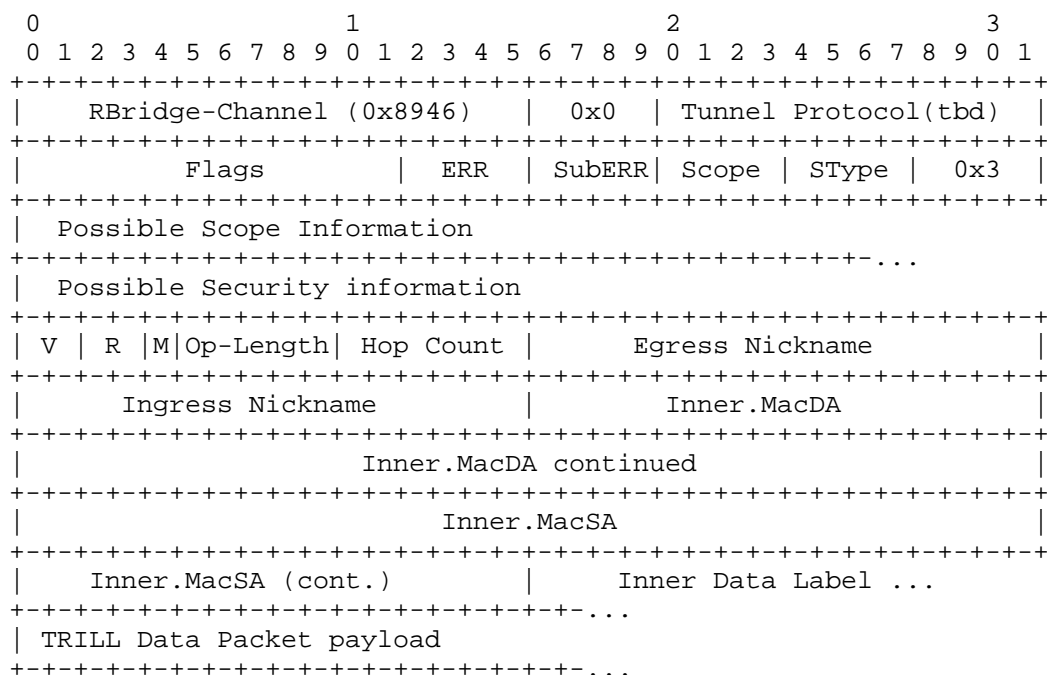


Figure 6. Nested TRILL Data Packet Channel Tunnel Structure

### 3.4 TRILL IS-IS Packet

A PType of 4 indicates that the payload of the Tunnel protocol message is an encapsulated TRILL IS-IS packet without the initial L2-IS-IS Ethertype as shown in the figure below. If this PType is implemented, the tunneled TRILL IS-IS packet is processed by the destination RBridge if it meets local policy. The intended use is to expedite the receipt of a link state PDU by some TRILL switch with an immediate requirement for the enclosed link state data. It is RECOMMENDED that any link local IS-IS PDU (Hello, xSNP, MTU-x) received via this channel tunnel payload type be discarded.

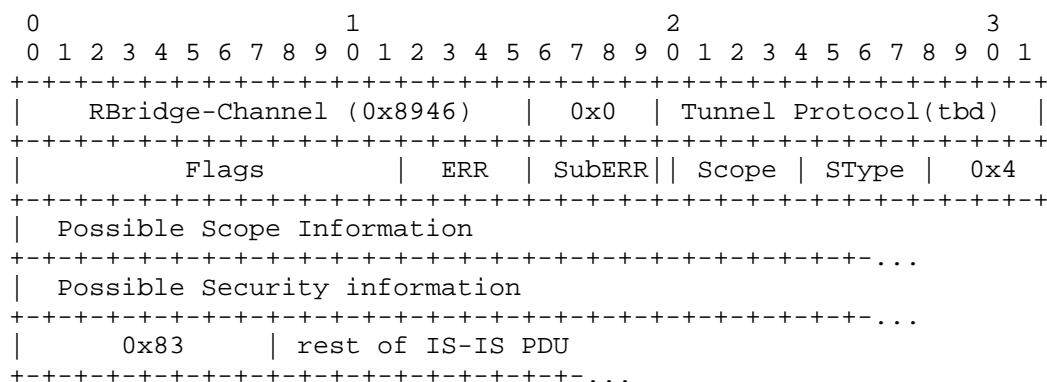


Figure 7. Tunneled TRILL IS-IS Packet Structure

### 3.5 Ethernet Frame

If PType is 5, the Tunnel Protocol payload is an Ethernet frame as might be received from or sent to an end station except that the tunneled Ethernet frame's FCS is omitted, as shown in Figure 8. (There is still an overall FCS if the RBridge Channel message is being sent on an Ethernet link.) If this PType is implemented, the tunneled frame is handled as if it had been received on the port on which the Tunnel Protocol message was received.

In the case of a non-Ethernet link, such as a PPP link [RFC6361], the ports on the link are considered to have link local synthetic 48-bit MAC addresses constructed by concatenating three 16-bit quantities: 0xFEFF, the primary nickname of the TRILL switch (see Section 1.2), and the Port ID that the RBridge has assigned to that port, as shown in Figure 9. The resulting MAC address has the Local bit on and the Group bit off [RFC5342bis]. Since end stations are connected to TRILL switches only over Ethernet, there can be no end stations on a non-Ethernet link in a TRILL campus. Thus such synthetic MAC addresses cannot conflict on the link with an end station address.

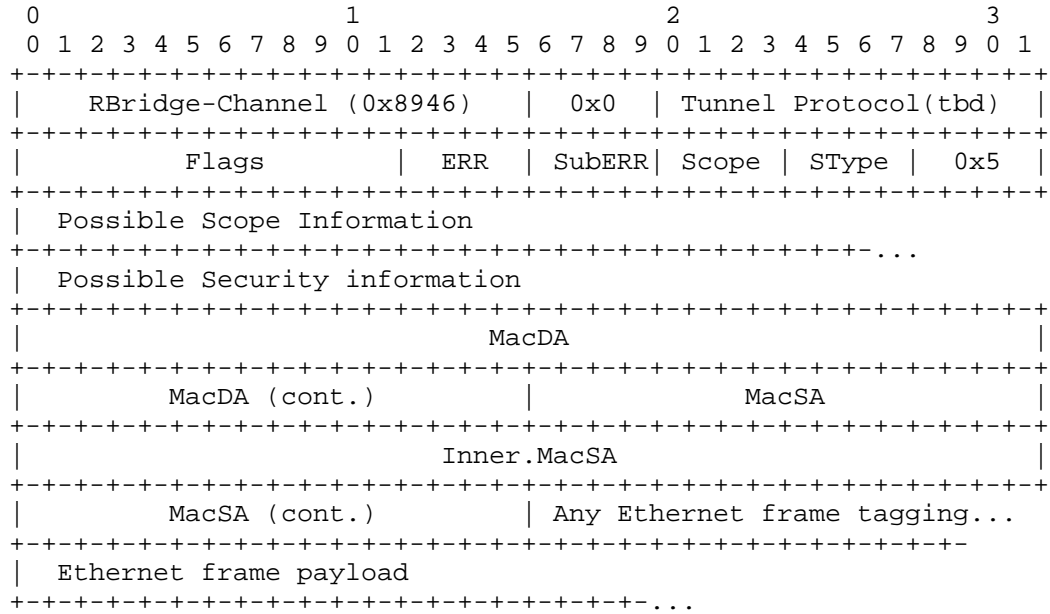


Figure 8. Ethernet Frame Channel Tunnel Structure

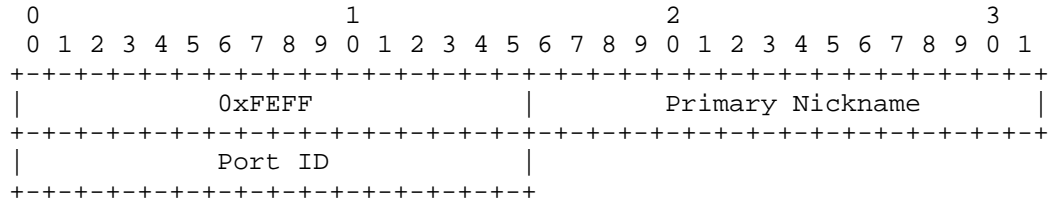


Figure 9. Synthetic MAC Address

### 3. Channel Tunnel Scopes

The Channel Tunnel protocol extends the RBridge Channel facility to optionally support typed messages between an end station and a TRILL switch, in either direction, or between two end stations, when these devices are part of the same TRILL campus but not on the same link. The scopes specified in this document are as follows:

Scope	Symbol	Section	From-To
0	NORM		Normal
1	ESRB	3.1	End Station to RBridge
2	RBES	3.2	RBridge to End Station
3	ESES	3.3	End Station to End Station
4-14		Available for assignment by IETF Review	
15		Reserved	

Table 2. Scope Values

If the Channel Tunnel protocol is supported, then the NORM scope MUST be supported. All other scopes MAY be supported. In cases where a sequence of steps is given, other processing sequences producing the same result are, as always, allowed. The detail are given below.

**NORM:** This is the normal scope of an RBridge Channel message. The base RBridge Channel mechanisms apply [RFCchannel]. The scope dependent addressing information is of zero length. This scope is typically used when just the security or payload type features of the Tunnel Protocol are desired. If a TRILL switch supports the Channel Tunnel facility, it MUST support NORM scope.

**ESRB:** From end station to RBridge(s) not on the same link. The scope dependent address information is eight bytes long. See Section 4.1 for further details. This scope MAY be supported.

**RBES:** From RBridge to end station not on the same link. The scope dependent address information is eight bytes long. See Section 4.2 for further details. This scope MAY be supported.

**ESES:** From end station to en station not on the same link. The scope information is twelve bytes long. See Section 4.3 for further details. This scope MAY be supported.

It is an implementation option and may depend on local policy whether or not an edge TRILL switch that has been requested to forward a Channel Tunnel protocol message due to a non-NORM Scope examines the SType and, if it does examine the SType, whether it verifies any authentication.



### 3.1 End Station to RBridge(s)

The ESRB scope additional information is as follows:

```

+-----+-----+
| Scope Destination Nickname | (2 bytes)
+-----+-----+
| Scope Source MAC Address   | (6 bytes)
+-----+-----+

```

Figure 10. ESRB Scope Information

To support the case where an end station originates a multi-destination RBridge Channel message to all the TRILL switches advertising interest in a Data Label, the BR (Broadcast) bit in the RBridge Channel Header Flags field is used (see Section 7).

Steps by the source end station:

If the RBridge Channel message is intended to a single destination RBridge, the source end station sets the Scope Destination Nickname to the nickname of that RBridge and ensures that the BR bit is zero. If the message is intended to be broadcast to the RBridges indicating interest in a Data Label, the end stations sets the BR bit, uses that Data Label as part of the TRILL Header information, and the contents of the Scope Destination Nickname field is ignored.

Steps by the ingress TRILL switch on receiving the native RBridge Channel message from the end station:

0. As with any RBridge Channel message, determine, as a matter of local policy, whether the native RBridge Channel message is acceptable and discard it if it is not. This test might take into account, for example, whether the message is authenticated (see Section 5), whether or not the BR flag is set, and whether or not the original native destination MAC address is All-Edge-RBridges.
1. Store the native RBridge Channel message's source MAC address into the Scope Source MAC Address field.
2. Clear the NA bit and set the MH bit in the RBridge Channel Header flags.
3. Set the native RBridge Channel message's MAC destination address to All-Egress-RBridges.
4. Set the native RBridge Channel message's MAC source address to the MAC address that the ingress RBridge normally uses as the Inner.MacSA for RBridge Channel messages it originates.
- 5.a. If the BR flag is zero, ingress the modified native frame as a unicast TRILL RBridge Channel message with egress nickname set from the Scope Destination Nickname. If that Scope

- Destination Nickname is unknown, the appropriate error SHOULD be returned (see Section 6).
- 5.b If the BR flag is one, select a distribution tree and ingress the modified native frame as a multi-destination TRILL RBridge Channel message.
  - 5.c Regardless of the BR flag value, the Inner.VLAN is the VLAN ID reported by the ingress port or, if that port is configured for FGL, the Inner.Lable is the FGL that VLAN maps to.
  6. Process the resulting RBridge Channel message. Note that if it is unicast to the ingress RBridge as egress, it is then egressed. And if it is multi-destination and the ingress RBridge qualifies, a copy is egressed as well as a copy being sent on the selected distribution tree.

#### 4.2 RBridge to End Station

The RBES scope additional addressing information is as follows:

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+...--+
| Scope Destination MAC Address | (6 bytes)
+-----+-----+-----+-----+-----+-----+-----+-----+-----+...--+
| Scope Source Nickname | (2 bytes)
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 11. RBES Scope Information

Steps by the source TRILL switch:

The source RBridge must set the Scope Destination MAC Address field. It creates an RBridge Channel message, either unicast or multi-destination, based on that MAC address. (The Inner.MacDA cannot be used for this because it must be the All-Egress-RBridges MAC address.) The created RBridge Channel message is unicast if the Scope Destination MAC address is unicast and the creating RBridge knows the egress to which that MAC address is connect. The created RBridge channel message is multi-destination if the Scope Destination MAC Address is broadcast, multicast, or unknown unicast. The source RBridge sets the Inner.MacSA to the MAC address it usually uses for RBridge Channel messages and also selects the Inner VLAN or FGL.

Steps by the egress TRILL switch(es):

The egress TRILL switch stores the ingress nickname into the Scope Source Nickname and sets the NA bit in the RBridge Channel Header flags. It then egresses the frame as a native RBridge Channel message, setting the native frame's outer destination and source MAC addresses to the Scope Destination MAC Address and the egress

RBridge port's MAC address, respectively.

If the original RBridge Channel message was multi-destination it might be egressed by more than one TRILL switch, each of which would perform the above transform. Whether such a multi-destination RBridge Channel Tunnel Protocol message would be accepted by any particular egress TRILL switch is a matter of local policy.

#### 4.3 End Station to End Station

The ESES scope additional addressing information is as follows:

```

+-----+
| Scope Destination MAC Address | (6 bytes)
+-----+
| Scope Source MAC Address      | (6 bytes)
+-----+

```

Figure 12. ESES Scope Information

Steps by the source end stations:

If the RBridge Channel message is intended for a single destination end station, the source end station sets the Scope Destination MAC address to the MAC address of that end station and ensures that the BR bit is zero. If the message is intended to be broadcast to a set of end stations via a multicast MAC address or the broadcast MAC address, the end station sets the Scope Destination MAC address to that multicast or broadcast address and sets the BR bit. All of this is within the VLAN of the native RBridge Channel message or its Fine Grained Label (FGL) if the ingress port is configured to map to an FGL.

Steps by the ingress TRILL switch:

0. As with any RBridge Channel message, determine, as a matter of local policy, whether the native RBridge Channel message is acceptable and discard it if it is not. This test might take into account, for example, whether the message is authenticated (see Section 5), whether or not the BR flag is set, and whether or not the original Outer.MacDA is All-Edge-RBridges.
1. Store the native RBridge Channel message's source MAC address into the Scope Source MAC Address.
2. Clear the NA bit and set the MH bit in the RBridge Channel Header flags.
3. Set the native RBridge Channel message's MAC destination address to All-Egress-RBridges.

4. Set the native RBridge Channel message's MAC source address to the MAC address that the ingress RBridge normally uses as the Inner.MacSA for RBridge Channel messages it originates.
- 5.a. If the BR flag is zero, lookup the Scope Destination MAC Address and ingress the modified native frame as if it were a unicast native frame with that destination MAC address. This will result in either a unicast TRILL Data packet to the Scope Destination MAC Address or in unknown MAC flooding.
- 5.b If the BR flag is one, select a distribution tree and ingress the modified native frame as a multi-destination TRILL RBridge Channel message.
- 5.c Regardless of the BR flag value, the Inner.VLAN is the VLAN ID reported by the ingress port or, if that port is configured for FGL, the Inner.Lable is the FGL that VLAN maps to.
6. Process the resulting RBridge Channel message. Note that if it is unicast to the ingress RBridge as egress, it is then egressed. And if it is multi-destination and the ingress RBridge qualifies, a copy is egressed as well as a copy being sent on the selected distribution tree. It is possible that the Scope Destination MAC is actually out a different or even the same port of the ingress TRILL switch as the port on which the native RBridge Channel message was received.

Steps by the egress TRILL switch(es):

The egress RBridge sets the NA bit in the RBridge Channel Header flags. It then egresses the frame as a native RBridge Channel message, setting the native frame's outer destination and source MAC addresses to the Scope Destination MAC Address and the egress RBridge port's MAC address, respectively.

If the original RBridge Channel message was multi-destination it might be egressed by more than one RBridge, each of which would perform the above transform. Whether such a multi-destination RBridge Channel Tunnel Protocol message would be accepted by egress RBridges is a matter of local policy.

## 5. Security, Keying, and Algorithms

The following table gives the assigned values of the SType field and their meaning.

SType	Section	Meaning
0	5.1	None
1	5.2	RFC 5310 Based Authentication
2	5.3	DTLS Based Security
3-14		Available for assignment on IETF Review
15		Reserved

Table 3. SType Values

For all SType values except zero, the Security Information starts with a byte of flag bits and a byte of remaining length as follows:

```

+-----+
|A|E|   RESV   |   Size   |   Info
+-----+

```

Figure 12. Security Information Format

The fields are as follows:

A: Zero if authentication is not being provided. One if it is.

E: Zero if encryption is not being provided. One if it is.

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

Size: The number of byte of Info as an unsigned byte.

Info: Variable length Security Information.

### 5.1 SType None

No security services are being invoked. The length of the Security Information field (see Figure 6) is zero.

### 5.2 RFC 5310 Based Authentication

The security information is the same as the value of the Authentication TLV as specified in [RFC5310]. See figure below.

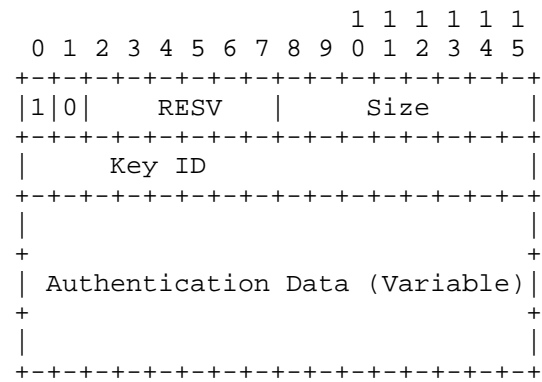


Figure 13. SType 1 Security Information

The Key ID normally specifies a keying value and algorithm.

5.3 DTLS Based Security

TBD - permits key negotiation, provides both encryption and authentication [RFC6347]...

## 6. Channel Tunnel Errors

RBridge Channel Tunnel Protocol errors are reported like RBridge Channel level errors. The ERR field is set to one of the following error codes:

ERR	Meaning
---	-----
6	Unknown or unsupported field value
7	Authentication failure
(more TBD)	

Table 4. Additional ERR Values

### 6.1 SubERRs under ERR 6

If the ERR field is 6, the SubERR field indicates

SubERR	Meaning (for ERR = 6)
-----	-----
0	Unsupported Scope
1	Unsupported SType
2	Unsupported PType
3	Unknown or reserved Scope Egress Nickname in an ESRB scope Tunnel Channel message.
4	Unsupported crypto algorithm
(more TBD)	

Table 5. SubERR values under ERR 6

## 7. IANA Considerations

IANA is requested to allocate a new RBridge Channel protocol number from the range based on Standards Action for the "Channel Tunnel" protocol.

IANA is requested to allocate a new RBridge Channel Header flag bit for the Broadcast (BR) flag with this document as reference.

## 8. Security Considerations

The RBridge Channel tunnel facility has potentially positive and negative effects on security.

On the positive side, it provides optional security that can be used to authenticate and/or encrypt channel messages. Some RBridge Channel message payloads provide their own security [RFCbfd] but where this is not true, careful consideration should be give to requiring use of the security features of the Tunnel Protocol.

On the negative side, the ability to tunnel various payload types and to tunnel them not just between TRILL switches but to and from end stations can increase risk unless precautions are taking. The processing of decapsulated Tunnel Protocol payloads is not a good place to be liberal in what you accept as the tunneling facility makes it easier for unexpected messages to pop up in unexpected places in a TRILL campus due to accidents or the actions of an adversary. Local policies should generally be strict and only process payload types required and then only with adequate authentication for the particular circumstances.

See [RFCchannel] for general RBridge Channel Security Considerations.

See [RFC6325] for general TRILL Security Considerations.



## Normative References

- [RFC2119] - Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5310] - Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.
- [RFC6325] - Perlman, R., D. Eastlake, D. Dutt, S. Gai, and A. Ghanwani, "RBridges: Base Protocol Specification", RFC 6325, July 2011.
- [RFC6347] - Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.
- [ClearCorrect] - Eastlake, D., M. Zhang, A. Ghanwani, V. Manral, A. Banerjee, "TRILL: Clarifications, Corrections, and Updates", draft-ietf-trill-clear-correct, in RFC Editor's queue.
- [RFCchannel] - D. Eastlake, V. Manral, Y. Li, S. Aldrin, D. Ward, "TRILL: RBridge Channel Support", draft-ietf-trill-rbridge-channel-08.txt, in RFC Editor's queue.
- [RFCfgl] - D. Eastlake, M. Zhang, P. Agarwal, R Perlman, D. Dutt, "TRILL: Fine-Grained Labeling", draft-ietf-trill-fine-labeling, in RFC Editor's queue.

## Informative References

- [RFC6361] - Carlson, J. and D. Eastlake 3rd, "PPP Transparent Interconnection of Lots of Links (TRILL) Protocol Control Protocol", RFC 6361, August 2011
- [RFC5342bis] - D. Eastlake, J. Abley, " IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", draft-eastlake-rfc5342bis, work in progress.
- [RFCbfd] - Manral, V., D. Eastlake, D. Ward, A. Banerjee, "TRILL (Transparent Interconnection of Lots of Links): Bidirectional Forwarding Detection (BFD) Support", draft-ietf-trill-rbridge-bfd, in RFC Editor's queue.

#### Acknowledgements

The contributions of the following are hereby acknowledged:

TBD

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

Authors' Addresses

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

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

Yizhou Li  
Huawei Technologies  
101 Software Avenue,  
Nanjing 210012, China

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

## Copyright, Disclaimer, and Additional IPR Provisions

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

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



INTERNET-DRAFT  
Intended status: Proposed Standard

Donald Eastlake  
Yizhou Li  
Huawei  
Radia Perlman  
Intel

Expires: April 19, 2014      October 20, 2013

TRILL: Interface Addresses APPsub-TLV  
<draft-eastlake-trill-ia-appsubtlv-03.txt>

#### Abstract

This document specifies a TRILL (Transparent Interconnection of Lots of Links) IS-IS application sub-TLV that enables the reporting by a TRILL switch of sets of addresses such that all of the addresses in each set designate the same interface (port). For example, an EUI-48 MAC (Extended Unique Identifier 48-bit, Media Access Control) address, IPv4 address, and IPv6 address can be reported as all corresponding to the same interface. Such information could be use in some cases to synthesize responses to or by-pass the need for the Address Resolution Protocol (ARP), the IPv6 Neighbor Discovery (ND) protocol, or the flooding of unknown MAC addresses.

#### Status of This Memo

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

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

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

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

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

## Table of Contents

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

## 1. Introduction

This document specifies a TRILL (Transparent Interconnection of Lots of Links) [RFC6325] IS-IS application sub-TLV (APPsub-TLV [RFC6823]) that enables the convenient representation of sets of addresses such that all of the addresses in each set designate the same interface (port). For example, an EUI-48 MAC (Extended Unique Identifier 48-bit, Media Access Control [RFC5342bis]) address, IPv4 address, and IPv6 address can be reported as all three designating the same interface. In addition, a Data Label (VLAN or Fine Grained Label (FGL [RFCfgl])) is specified for the interface along with the TRILL switch and, optional the TRILL switch port, from which the interface is reachable. Such information could be use in some cases to synthesize responses to or by-pass the need for the Address Resolution Protocol (ARP [RFC826]), the IPv6 Neighbor Discovery (ND [RFC4861]) protocol, or the flooding of unknown MAC addresses [DirectoryFramework].

This APPsub-TLV appears inside the TRILL GENINFO TLV specified in [ESADI] but may also occur in other application contexts. Directory Assisted TRILL Edge services [DirectoryScheme] are expected to make use of this APPsub-TLV.

Although, in some IETF protocols, address field types are represented by Ethertype [RFC5342bis] or Hardware Type [RFC5494], only Address Family Number (AFN) is used in this APPsub-TLV to represent address field type.

### 1.1 Conventions Used in This Document

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

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

AFN: Address Family Number

APPsub-TLV: Application sub-TLV [RFC6823].

Data Label: VLAN or FGL.

FGL: Fine Grained Label [RFCfgl].

IA: Interface Addresses.

RBridge: An alternative name for a TRILL switch.



TRILL switch: A device that implements the TRILL protocol.

The Interface Addresses (IA) APPsub-TLV is used to advertise that a set of addresses indicate the same interface (port) within a Data Label (VLAN or FGL) and to associate that interface with the TRILL switch, and optionally the TRILL switch port, by which the interface is reachable. These addresses can be in different address families. For example, it can be used to declare that a particular interface with specified IPv4, IPv6, and EUI-48 MAC addresses in some particular Data Label is reachable from a particular TRILL switch.

A device or application making use of IA APPsub-TLV data is not required to make use of all IA data. For example, a device or application that was only interested in MAC and IPv6 addresses could ignore any IPv4 or other types of address information that was present.

[Page 5]

Figure 1. The Interface Addresses APPsub-TLV

- o Type: Interface Addresses TRILL APPsub-TLV type, set to TBD[#2 suggested] (IA-SUBTLV).
- o Length: Variable, minimum 6, maximum 250 when inside a TRILL GENINFO TLV [ESADI], maximum 255 in unconstrained contexts. If length is 5 or less or if the APPsub-TLV extends beyond an encompassing TRILL GENINFO TLV, the APPsub-TLV MUST be ignored.
- o Nickname: The nickname of the TRILL switch by which the address sets are reachable. If zero, the address sets are reachable from the TRILL switch originating the message containing the APPsub-TLV (for example, an [ESADI] message).
- o Flags: A byte of flags as follows:

```

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

```

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

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

N: Notify flag: When a TRILL switch receives a new IA APPsub-TLV (one in a ESADI LSP fragment with a higher sequence number or a new message of some other type) and the N bit is one, the TRILL switch then checks the contents of the APPsub-TLV for IP address to MAC address mappings. If an IPv4 to MAC address mapping is found, gratuitous ARPs [RFC826] are sent and if an IPv6 to MAC address mapping is found, spontaneous Neighbor Advertisements [RFC4861] are sent. In both cases, these are sent out all the ports of the TRILL switch that offer end station service and are in the VLAN or FGL of the APPsub-TLV information.

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

- o Confidence: This 8-bit unsigned quantity in the range 0 to 254 indicates the confidence level in the addresses being transported [RFC6325]. A value of 255 is treated as if it was 254.
- o Addr Sets End: The unsigned offset of the byte, within the IA APPsub-TLV value part, of the last byte of the last Address Set.

This will be the byte just before the first sub-sub-TLV if any sub-sub-TLVs are present (see Section 3). If this is equal to Size, there are no sub-sub-TLVs. If this is greater than Size, the IA APPsub-TLV is corrupt and MUST be discarded.

- o Template: The initial byte of this field is the unsigned integer K. If K has a value from 1 to 31, it indicates that this initial byte is followed by a list of K AFNs (Address Family Numbers) that specify the exact structure and order of each Address Set occurring later in the APPsub-TLV. K can be 1, which is the minimum valid value. If K is zero, the IA APPsub-TLV is ignored. If K is 32 to 254, the length of the Template field is one byte and its value is intended to correspond to a particular ordered set of AFNs some of which are specified below. If K is 255, the length of the Template field is three bytes and the values of the second and third byte, considered as an unsigned integer in network byte order, are reserved to correspond to future specified ordered sets of AFNs.

If the Template uses explicit AFNs, it looks like the following.

```

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

```

For K in the 32 to 103 range, values indicate combinations of a specific number of MAC addresses, IPv4 addresses, IPv6 addresses, and TRILL switch port IDs in that order. The value of K is

$$K = 32 + M + 3*v4 + 9*v6 + 36*P$$

where M is 0, 1, or 2 (0 if no MAC address is present, 1 if a 48-bit MAC is present, 2 if a MAC/24 (see Section 5.1) is present), v4 is the number of IPv4 addresses (limited to 0, 1, or 2) and v6 is the number of IPv6 addresses (limited to 0 through 3 inclusive), and P is the number of TRILL switch port IDs (limited to 0 or 1). That equation specifies values of K from 32 through 103. Values from 104 through 254 of the byte value are available for assignment by Expert Review (see Section 5). K = 255 indicates a three byte Template field as specified above. All values (0 through 65,545) of this two byte value are available for assignment by Expert Review.

If an unknown Template K value in the range 104 to 254 is received or a K of 255 followed by an unknown two byte value, the IA APPsub-TLV MUST be ignored.

- o AFN: A two-byte Address Family Number. The number of AFNs present is given by K. There are no AFNs if K is greater than 31. The AFN sequence specifies the structure of the Address Sets occurring later in the TLV. For example, if Template Size is 2 and the two AFNs present are the AFNs for EUI-48 and IPv4, in that order, then each Address set present will consist of a 6-byte MAC address followed by a 4-byte IPv4 address. If any AFNs are present that are unknown to the receiving IS and the length of the corresponding address is not provided by a sub-sub-TLV as specified below, the receiving IS will be unable to parse the Address Sets and MUST ignore the IA APPsub-TLV.
- o Address Set: Each address set in the APPsub-TLV consists of exactly the same sequence of addresses of the types specified by the Template earlier in the APPsub-TLV. No alignment, other than to a byte boundary, is guaranteed. The addresses in each Address Set are contiguous with no unused bytes between them and the Address Sets are contiguous with no unused bytes between successive Address Sets. The Address Sets must fit within the TLV. If the product of the size of an Address Set and the number of Address Sets is so large that this is not true, the IA APPsub-TLV is ignored.
- o sub-sub-TLVs: If the Address Sets indicated by Addr Sets End do not completely fill the Length of the APPsub-TLV, the remaining bytes are parsed as sub-sub-TLVs [RFC5305]. Any such sub-sub-TLVs that are not known to the receiving RBridge are ignored. Should this parsing not be possible, for example there is only one remaining byte or an apparent sub-sub-TLV extends beyond the end of the TLV, the containing IA APPsub-TLV is considered corrupt and is ignored. (Several sub-sub-TLV types are specified in Section 3.)

Different IA APPsub-TLVs within the same or different LSPs or other data structures may have different Templates. The same AFN may occur more than once in a Template and the same address may occur in different address sets. For example, an EUI-48 MAC address interface might have three different IPv6 addresses. This could be represented by an IA APPsub-TLV whose Template specifically provided for one EUI-48 address and three IPv6 addresses, which might be an efficient format if there were multiple interfaces with that pattern. Alternatively, a Template with one EUI-48 and one IPv6 address could be used in an IA APPsub-TLV with three address sets each having the same EUI-48 address but different IPv6 addresses, which might be the most efficient format if only one interface had multiple IPv6 addresses and other interfaces had only one IPv6 address.

In order to be able to parse the Address Sets, a receiving RBridge must know at least the size of the address each AFN the Template specifies; however, the presence of the Addr Set End field means that the sub-sub-TLVs, if any, can always be located by a receiver. An RBridge can be assumed to know the size of the AFNs mentioned in Section 5. Should an RBridge wish to include an AFN that some receiving RBridge in the campus may not know, it SHOULD include an AFN-Size sub-sub-TLV as described below. If an IA APPsub-TLV is received with one or more AFNs in its template for which the receiving RBridge does not know the length and for which an AFN-Size sub-sub-TLV is not present, that IA APPsub-TLV MUST be ignored.

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

IA APPsub-TLVs can have trailing sub-sub-TLVs [RFC5305] as specified below. These sub-sub-TLVs occur after the Address Sets and the amount of space available for sub-sub-TLVs is determined from the overall IA APPsub-TLV length and the value of the Addr Set End byte.

There is no ordering restriction on sub-sub-TLVs. Unless otherwise specified each sub-sub-TLV type can occur zero, one, or many times in an IA APPsub-TLV.

#### 3.1 AFN Size sub-sub-TLV

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

1. One or more AFNs that are unknown to the receiving RBridge appears in the template. If an AFN Size sub-sub-TLV is present for each such AFN, then at least the IA APPsub-TLV can be parsed and possibly other addresses in each address set can still be used.
2. If an AFN occurs in the Template that represents a variable length address, this sub-sub-TLV gives its size for all occurrences in that IA APPsub-TLV. (It is believed that the addresses specified by all currently assigned AFNs are fixed length.)

```

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

Where each AFN Size Record is structured as follows:

```

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

- o Type: AFN-Size sub-sub-TLV type, set to 1 (AFNsz).
- o Length: 3\*n where n is the number of AFN Size Records present. If Length is not a multiple of 3, the sub-sub-TLV MUST be ignored.

- o AFN Size Record(s): Zero or more 3-byte records, each giving the size of an address type identified by an AFN,
- o AFN: The AFN whose length is being specified by the AFN Size Record.
- o AdrSize: The length in bytes of addresses specified by the AFN field as an unsigned integer.

An AFN Size sub-sub-TLV for any AFN known to the receiving RBridge is compared with the size known to the RBridge. If they differ the IA APPsub-TLV is assumed to be corrupt and MUST be ignored.

### 3.2 Fixed Address sub-sub-TLV

There may be cases where, in an Interface Addresses APP-subTLV, the same address would appear in every address set across the APP-subTLV. To avoid wasted space, this sub-sub-TLV can be used to indicate such a fixed address. The address or addresses incorporated into the sets by this sub-sub-TLV are NOT mentioned in the IA APPsub-TLV Template.

```

+---+---+---+---+---+
| Type=FIXEDADR |           (1 byte)
+---+---+---+---+---+
| Length       |           (1 byte)
+---+---+---+---+---+
| AFN          |           (2 bytes)
+---+---+---+---+---+...
| Fixed Address |           (variable)
+---+---+---+---+---+...
```

- o Type: Data Label sub-sub-TLV type, set to 2 (FIXEDADR).
- o Length: variable, minimum 3. If Length is 2 or less, the sub-sub-TLV MUST be ignored.
- o AFN: Address Family Number of the Fixed Address.
- o Fixed Address: The address of the type indicated by the preceding AFN field that is considered to be part of every Address Set in the IA APPsub-TLV.

### 3.3 Data Label sub-sub-TLV

This sub-sub-TLV indicates the Data Label within which the interfaces listed in the IA APPsub-TLV are reachable. It is useful if the IA



APPsub-TLV occurs outside of the context of an [ESADI] or other type of message specifying the Data Label or if it is desired and permitted to override that specification. Multiple occurrences of this sub-sub-TLV indicate that the interface is reachable in all of the Data Labels given.

```

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

```

- o Type: Data Label sub-TLV type, set to 3 (LABEL).
- o Length: 2 or 3. If Length is some other value, the sub-sub-TLV is ignored.
- o Data Label: If length is 2, the bottom 12 bits of the Data Label are a VLAN ID and the top 4 bits are reserved (MUST be sent as zero and ignored on receipt). If the length is 3, the three Data Label bytes contain an FGL [RFCfgl].

### 3.4 Topology sub-sub-TLV

The presence of this sub-sub-TLV indicates that the interfaces given in the IA APPsub-TLV are reachable in the topology give. It is useful if the IA APPsub-TLV occurs outside of the context of an [ESADI] or other type of message indicating the topology or if it is desired and permitted to override that specification. If it occurs multiple times, then the Address Sets are in all of the topologies given.

```

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

```

- o Type: Topology sub-TLV type, set to 4 (TOPOLOGY).
- o Length: 2. If Length is some other values, the sub-sub-TLV is ignored.

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

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

#### 4. Security Considerations

The integrity of address mapping and reachability information and the correctness of Data Labels (VLANs or FLGs [RFCfgl]) are very important. Forged, altered, or incorrect address mapping or Data Labeling can lead to delivery of packets to the incorrect party, violating security policy. However, this document merely describes a data format and does not provide any explicit mechanisms for securing that information, other than a few trivial consistency checks that might detect some corrupted data. Security on the wire, or in storage, for this data is to be providing by the transport or storage used. For example, when transported with [ESADI], [ESADI] security mechanisms can be used.

The address mapping and reachability information, if known to be complete and correct, can be used to detect some cases of forged packet source addresses [DirectoryFramework]. In particular, if native traffic is received by a TRILL switch that would otherwise accept it but authoritative data indicates the source address should not be reachable from the receiving TRILL switch, that traffic should be discarded. The data format specified in this document may optionally include RBridge Port ID number so that this forged address filtering can be optionally applied with port granularity.

See [RFC6325] for general TRILL Security Considerations.

## 5. IANA Considerations

As specified below, IANA has allocated new AFN numbers and IANA is requested create the TRILL IS-APPsub-TLV sub-sub-TLV subregistry.

### 5.1 Additional AFN Number Allocation

IANA has allocated AFN numbers as follows:

Number	Description	References
-----	-----	-----
16391	OUI	This document.
16392	MAC/24	This document.
16393	MAC/40	This document.
16394	IPv6/64	This document.
16395	RBridge Port ID	This document.

The OUI AFN is provided so that MAC addresses can be abbreviated if they have the same upper 24 bits. In particular, if there is an OUI provided as a Fixed Address sub-sub-TLV (see Section 5.2.2) then, whenever a MAC/24 or MAC/40 address appears within an Address Set (as indicated by the Template), the OUI is used as the first 24 bits of the actual MAC address for the Address Set. An OUI provided by a Fixed Address sub-sub-TLV is ignored if the IA APPsub-TLV has no MAC/24 or MAC/40 in its template.

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

MAC/40 is a suffix as specified above except that it is 40-bit so the result of combining it with an OUI is a 64-bit MAC address.

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

Other AFNs can be found at <http://www.iana.org/assignments/address-family-numbers>

The following already allocated AFN values may be particularly useful for IA APPsub-TLVs:

Hex	Decimal	Description	References
-----	-----	-----	-----
0001	1	IPv4	
0002	2	IPv6	
4005	16,389	48-bit MAC	[RFC5342bis]
4006	16,390	64-bit MAC	[RFC5342bis]

## 5.2 IA APPsub-TLV Sub-Sub-TLVs SubRegistry

IANA is requested to establish a new subregistry of the TRILL Parameter Registry for sub-sub-TLVs of the Interface Addresses APPsub-TLV with initial contents as shown below.

Name: Interface Addresses APPsub-TLV Sub-Sub-TLVs

Procedure: Expert Review

Reference: This document

Type	Description	Reference
----	-----	-----
0	Reserved	
1	AFN Size	This document
2	Fixed Address	This document
3	Data Label	This document
4	Topology	This document
5-254	Available	
255	Reserved	

#### Acknowledgments

The authors gratefully acknowledge the contributions and review by the following:

Linda Dunbar

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

## Appendix A: Examples

Below are example IA APPsub-TLVs.

### A.1 Simple Example

Below is an annotated IA APPsub-TLV carrying two simple pairs of EUI-48 MAC addresses and IPv4 addresses from a Push Directory [DirectoryFramework]. No sub-sub-TLVs are included.

```

0x02(TBD)      Type: Interface Addresses
26             Size: 26 (=0x1A)
0x1234         RBridge Nickname from which reachable
0b100000000    Flags: Push Directory data
0xE3          Confidence
26            Address Sets End: 26 (=0x1A)
35            Template: 35 (0x23) = 32 + 1(MAC48) + 3*1(IPv4)

```

#### Address Set One

```

0x00005E0053A9  48-bit MAC address
198.51.100.23   IPv4 address

```

#### Address Set Two

```

0x00005E00536B  48-bit MAC address
203.0.113.201   IPv4 address

```

Size includes 6 for the fixed fields though and including the one byte template, plus 2 times the Address Set size. Each Address Set is 10 bytes, 6 for the 48-bit MAC address plus 4 for the IPv4 address. So total size is  $6 + 2*10 = 26$ .

See Section 2 for more information on Template.

### A.2 Complex Example

Below is an annotated IA APPsub-TLV carrying three sets of addresses, each consisting of an EUI-48 MAC address, an IPv4 addresses, an IPv6 address, and an RBridge Port ID, all from a Push Directory [DirectoryFramework]. The IPv6 address for each address set is synthesized from the MAC address given in that set and the IPv6/64 64-bit prefix provided through a Fixed Address sub-sub-TLV. In addition, a sub-sub-TLV is included that provides an FGL which overrides whatever Data Label may be provided by the envelope (for example [ESADI]) within which this IA APPsub-TLV occurs.

```

0x02(TBD)      Type: Interface Addresses
59             Size: 59 (=0x3B)
0x4321         RBridge Nickname from which reachable
0b100000000    Flags: Push Directory data
0xD3          Confidence
42            Address Sets End: 42 (=0x2A)
72            Template: 72(0x48)=32+1(MAC48)+3*1(IPv4)+36*1(P)

```

## Address Set One

```

0x00005E0053DE 48-bit MAC address
198.51.100.105  IPv4 address
0x1DE3         RBridge Port ID

```

## Address Set Two

```

0x00005E0053E3 48-bit MAC address
203.0.113.89   IPv4 address
0x1DEE         RBridge Port ID

```

## Address Set Three

```

0x00005E0053D3 48-bit MAC address
192.0.2.139    IPv4 address
0x01DE         RBridge Port ID

```

## sub-sub-TLV One

```

0x03          Type: Data Label
0x03          Length: implies FGL
0xD3E3E3     Fine Grained Label

```

## sub-sub-TLV Two

```

0x02          Type: Fixed Address
0x0A          Size: 0x0A = 10
0x400A        AFN: IPv6/64
0x0x20010DB800000000 IPv6 Prefix: 2001:DB8::

```

See Section 2 for more information on Template.

The Fixed Address sub-sub-TLV causes the IPv6/64 value give to be treated as if it occurred as a 4th entry inside each of the three Address Sets. When there is an IPv6/64 entry and a 48-bit MAC entry, the MAC value is expanded by inserting 0xFFFFE immediately after the OUI and the resulting 64-bit value is used as the lower 64 bits of the resulting IPv6 address [RFC5342bis]. As a result, a receiving TRILL switch would treat the three Address Sets shown as if they had an IPv6 address in them as follows:



Address Set One  
 0x20010DB800000000000005EFFFFE0053DE IPv6 Address

Address Set Two  
 0x20010DB800000000000005EFFFFE0053E3 IPv6 Address

Address Set Three  
 0x20010DB800000000000005EFFFFE0053D3 IPv6 Address

As an alternative to the compact "well know value" Template encoding used in this example above, the less compact explicit AFN encoding could have been used. In that case, the IA APPsub-TLV would have started as follows:

0x02(TBD)	Type: Interface Addresses
65	Size: 65 (=0x41)
0x4321	RBridge Nickname from which reachable
0b10000000	Flags: Push Directory data
0xD3	Confidence
48	Address Sets End: 48 (=0x30)
0x3	Template: 3 AFNs
0x4005	AFN: 48-bit MAC
0x0001	AFN: IPv4
0x400B	AFN: RBridge Port ID

As a final point, since the 48-bit MAC addresses in these three Address Sets all have the same OUI (the IANA OUI [RFC5342bis]), it would have been possible to just have a MAC/24 value giving the lower 24 bits of the MAC in each Address Set. The OUI would then be supplied by a second Fixed Address sub-sub-TLV providing the OUI. With N Address Sets, this would have saved 3\*N or 9 bytes in this case at the cost of 7 bytes (1 each for the type and length of the sub-sub-TLV, 2 for the OUI AFN number, and 3 for the OUI). So, even with just three Address Sets, there would be a small net saving of 2 bytes. The savings would grow with a larger number of Address Sets.

## Normative References

- [RFC826] Plummer, D., "An Ethernet Address Resolution Protocol", RFC 826, November 1982.
- [RFC2119] - Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997
- [RFC4861] - Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC5120] - Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.
- [RFC5305] - Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5342bis] - Eastlake 3rd, D., "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters", BCP 141, RFC 5342, September 2008.
- [RFC6325] - Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.
- [RFC6823] - Ginsberg, L., Previdi, S., and M. Shand, "Advertising Generic Information in IS-IS", RFC 6823, December 2012.
- [RFCfgl] - D. Eastlake, M. Zhang, P. Agarwal, R. Perlman, D. Dutt, "TRILL: Fine-Grained Labeling", draft-ietf-trill-fine-labeling-07.txt, in RFC Editor's queue.

## Informational References

- [ARP reduction] - Shah, et. al., "ARP Broadcast Reduction for Large Data Centers", Oct 2010.
- [DirectoryFramework] - Dunbar, L., D. Eastlake, R. Perlman, I. Gashinsky, "TRILL Edge Directory Assistance Framework", draft-ietf-trill-directory-framework-07.txt, in RFC Editor's queue.
- [DirectoryScheme] - Dunbar, L., D. Eastlake, R. Perlman, I. Gashinsky, Y. Li, "TRILL: Directory Assistance Mechanisms", draft-dunbar-trill-scheme-for-directory-assist, work in progress.

[ESADI] - Zhai, H., F. Hu, R. Perlman, D. Eastlake, O. Stokes, "TRILL (Transparent Interconnection of Lots of Links): The ESADI (End Station Address Distribution Information) Protocol", draft-ietf-trill-esadi-03.txt, work in progress.

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

Authors' Addresses

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

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

Yizhou Li  
Huawei Technologies  
101 Software Avenue,  
Nanjing 210012 China

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

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

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

## Copyright, Disclaimer, and Additional IPR Provisions

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

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



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

T. Mizrahi  
Marvell  
T. Senevirathne  
S. Salam  
D. Kumar  
Cisco  
D. Eastlake 3rd  
Huawei  
October 10, 2013

Loss and Delay Measurement in  
Transparent Interconnection of Lots of Links (TRILL)  
<draft-ietf-trill-loss-delay-00.txt>

Abstract

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

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on April 10, 2014.

## Copyright Notice

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

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

## Table of Contents

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



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

## 1. Introduction

TRILL [RFC2111] is a protocol for transparent least cost routing, where Rbridges route traffic to their destination based on least cost, using a TRILL encapsulation header with a hop count.

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

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

This document defines protocols for loss measurement and for delay measurement in TRILL networks. These protocols are somewhat based on the mechanisms defined in ITU-T G.8013/Y.1731 [Y.1731].

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

## 2. Conventions Used in this Document

### 2.1. Keywords

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

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

### 2.2. Definitions

- o One-way packet delay - (as defined in [OAM-REQ]) the time elapsed from the start of transmission of the first bit of a packet by an RBridge until the reception of the last bit of the packet by the remote RBridge.
- o Two-way packet delay - (as defined in [OAM-REQ]) the time elapsed from the start of transmission of the first bit of a packet from the local RBridge, receipt of the packet at the remote RBridge, the remote RBridge sending a response packet back to the local RBridge and the local RBridge receiving the last bit of that response packet.
- o Packet loss - the number of packets lost in a specific probe instance, and a specific observation period.

- o Far-end packet loss - the number of packets lost on the path from the local RBridge to the remote RBridge in a specific probe instance, and a specific observation period.
- o Near-end packet loss - the number of packets lost on the path from the remote RBridge to the local RBridge in a specific probe instance, and a specific observation period.

### 2.3. Abbreviations

1DM	One-way Delay Measurement message
1SL	One-way Synthetic Loss Measurement message
DMM	Delay Measurement Message
DMR	Delay Measurement Reply
MD	Maintenance Domain
MD-L	Maintenance Domain Level
MEP	Maintenance End Point
MIP	Maintenance Intermediate Point
MP	Maintenance Point
OAM	Operations, Administration and Maintenance
PM	Performance Monitoring
SLM	Synthetic Loss Measurement Message
SLR	Synthetic Loss Measurement Reply
TLV	Type, Length and Value
TRILL	Transparent Interconnection of Lots of Links

### 3. Loss and Delay Measurement in the TRILL Architecture

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

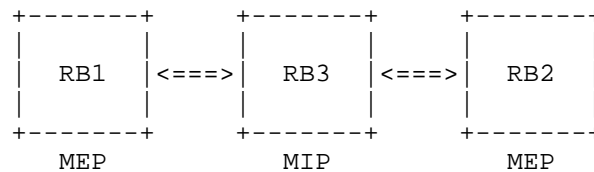


Figure 1 Maintenance Points in a TRILL Campus

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

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

### 3.1. Performance Monitoring Granularity

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

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

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

Paths in a TRILL network are not necessarily symmetric, i.e., a packet sent from RB1 to RB2 does not necessarily traverse the same set of RBridges or links as a packet sent from RB2 to RB1. Even within a given flow, packets from RB1 to RB2 do not necessarily traverse the same path as packets from RB2 to RB1. Therefore, this document provides tools for one-way performance monitoring and for two-way performance monitoring.

### 3.2.1. One-Way Performance Monitoring

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

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

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

### 3.2.2. Two-Way Performance Monitoring

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

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

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

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

As described in Section 3.1. , flow-based PM uses the Flow Entropy field as one of the parameters that identify a flow. In two-way PM, the Flow Entropy of the path from RB1 to RB2 is typically different from the Flow Entropy of the path from RB2 to RB1. This document uses the Reflector Entropy TLV [TRILL-FM]), which allows the sender to specify the Flow Entropy value to be used in the response message.

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

### 3.3. Point-to-point vs. Point-to-multipoint Performance Monitoring

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

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

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

## 4. Loss Measurement

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

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

### 4.1. One-Way Loss Measurement

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

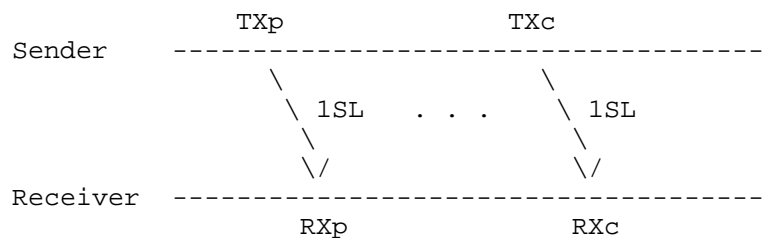


Figure 2 One-Way Loss Measurement

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

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

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

#### 4.1.1.1. 1SL Message Transmission

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

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

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

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

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

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

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

#### 4.1.2. 1SL Message Reception

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

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

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

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

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

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

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

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

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



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

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

#### 4.2. Two-Way Loss Measurement

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

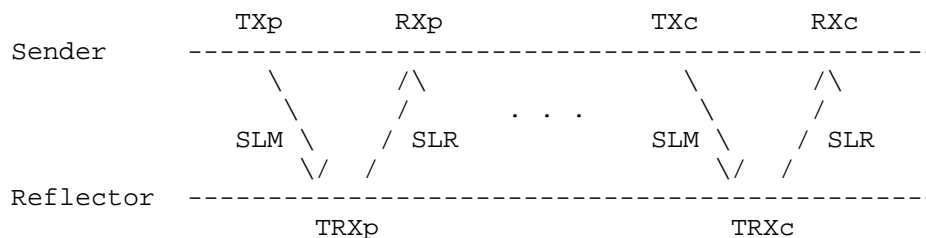


Figure 3 Two-Way Loss Measurement

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

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

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

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

#### 4.2.1. SLM Message Transmission

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

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

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

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

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

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

#### 4.2.2. SLM Message Reception

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

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

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

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

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

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

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

#### 4.2.3. SLR Message Reception

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

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

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

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

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

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

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

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

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

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

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

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

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

## 5. Delay Measurement

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

### 5.1. One-Way Delay Measurement

One-way Delay Measurement is used for computing the one-way packet delay from one MEP to another. The packet format used in one-way Delay Measurement is referred to as 1DM, and is specified in Section

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

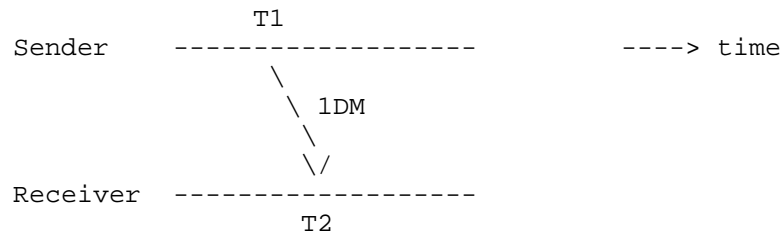


Figure 4 One-Way Delay Measurement

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

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

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

#### 5.1.1.1. 1DM Message Transmission

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

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

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

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

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

### 5.1.2. 1DM Message Reception

Upon receiving a 1DM packet, the receiver records its time of reception, T2. The receiver MUST verify two conditions:

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

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

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

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

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

### 5.2. Two-Way Delay Measurement

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

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

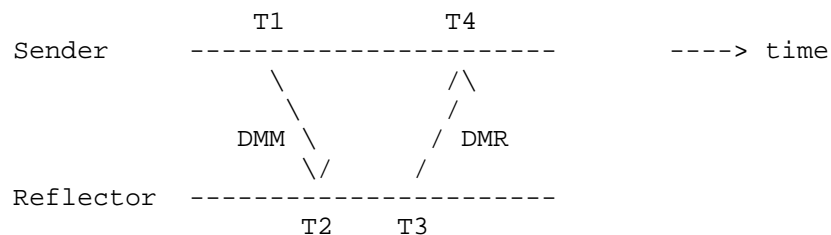


Figure 5 Two-Way Delay Measurement

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

#### 5.2.1. DMM Message Transmission

DMM packets can be transmitted periodically or on-demand.

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

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

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

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

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

#### 5.2.2. DMM Message Reception

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

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

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

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

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

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

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

#### 5.2.3. DMR Message Reception

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

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

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

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

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

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

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

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

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



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

## 6. Packet Formats

### 6.1. TRILL OAM Encapsulation

The TRILL OAM encapsulation is defined in [OAM-FRAMEWK], and is quoted in this document for clarity. For further details see [OAM-FRAMEWK].

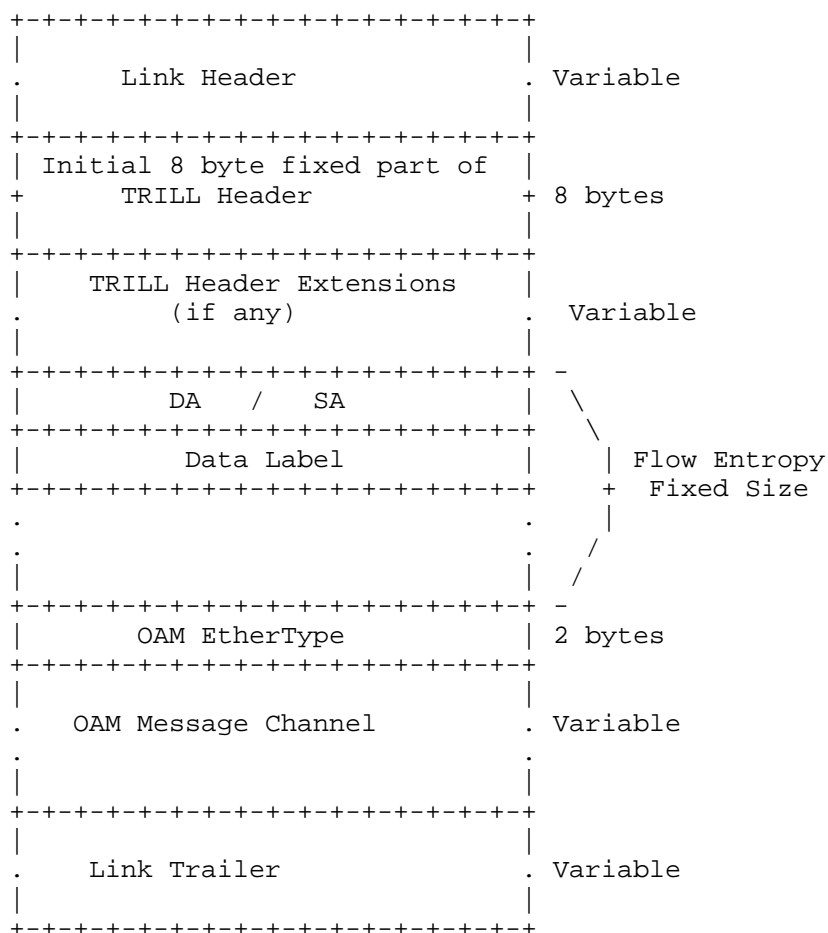


Figure 6 TRILL OAM Encapsulation

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

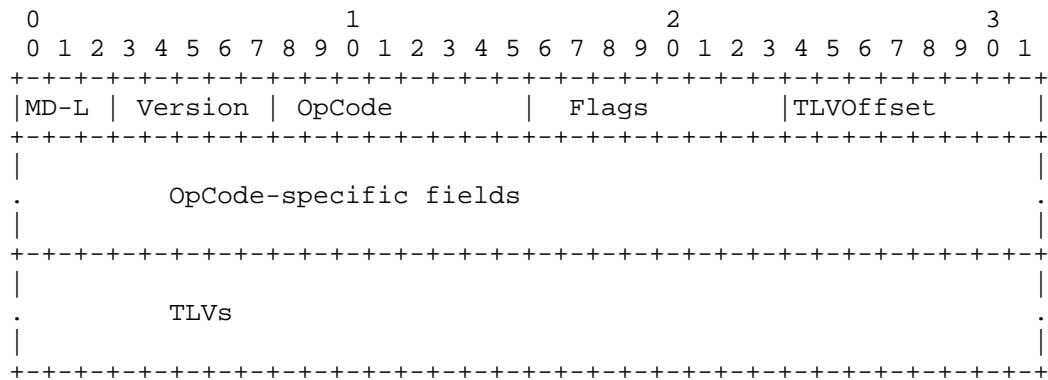


Figure 7 OAM Packet Format

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

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

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

## 6.2. Loss Measurement Packet Formats

### 6.2.1. Counter Format

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

## 6.2.2. 1SL Packet Format

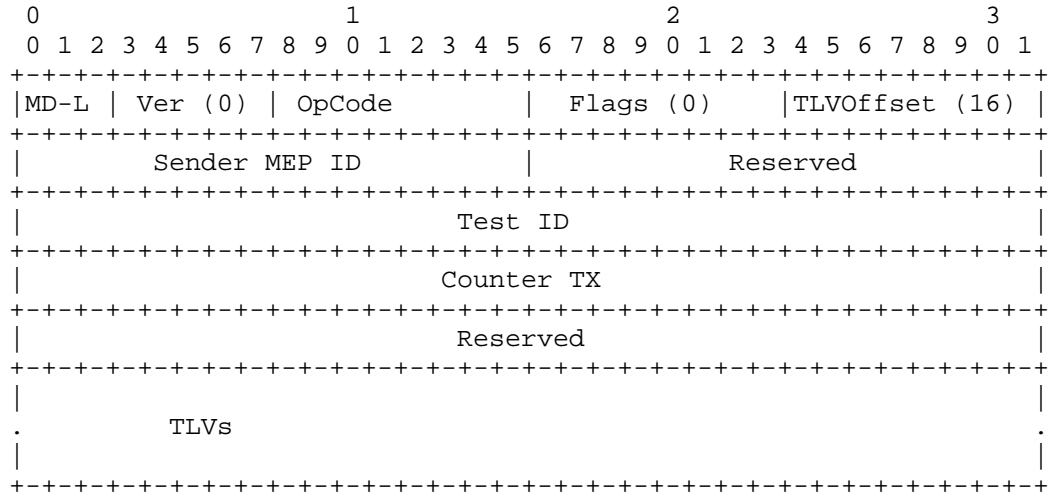


Figure 8 1SL Packet Format

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

## 6.2.3. SLM Packet Format

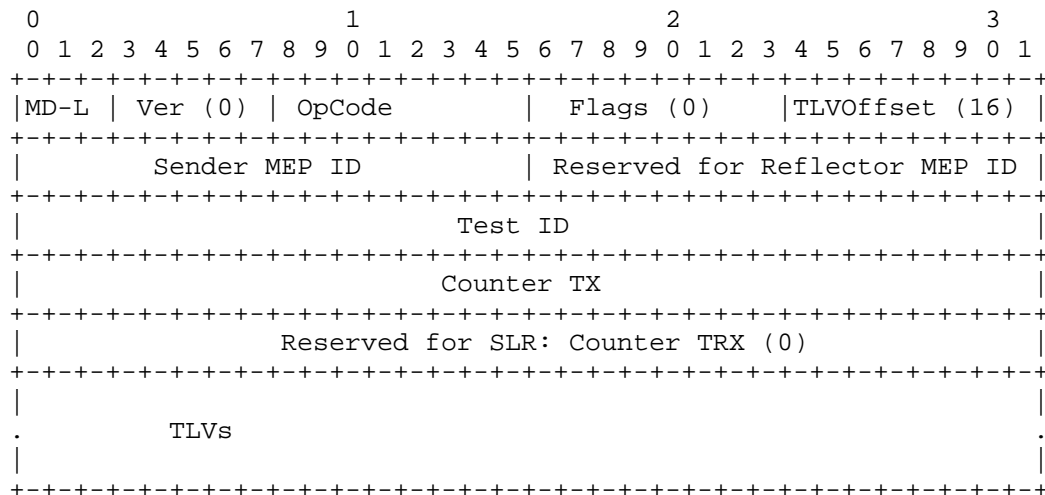


Figure 9 SLM Packet Format

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

## 6.2.4. SLR Packet Format

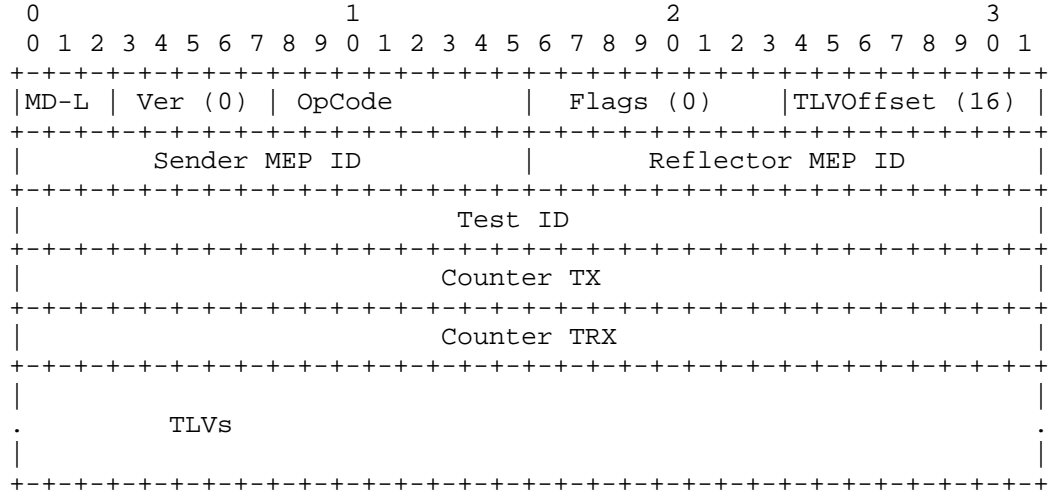


Figure 10 SLR Packet Format

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

### 6.3. Delay Measurement Packet Formats

#### 6.3.1. Timestamp Format

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

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

#### 6.3.2. 1DM Packet Format

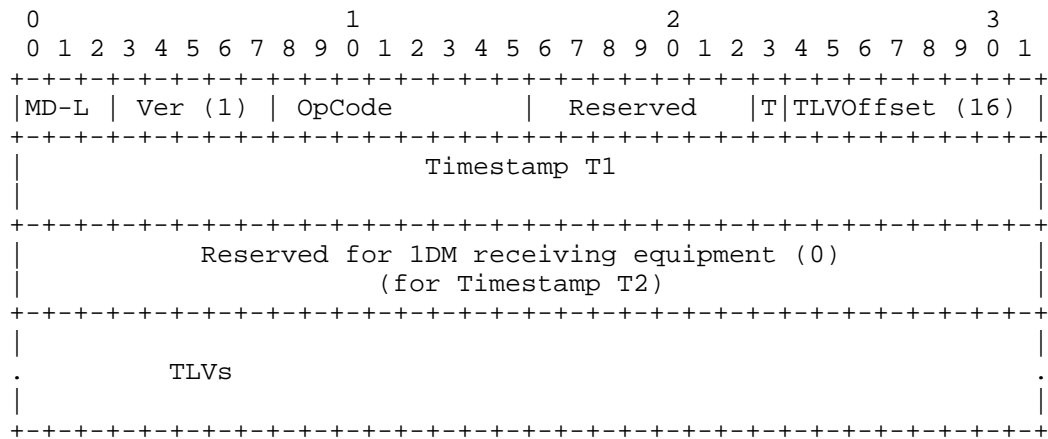


Figure 11 1DM Packet Format

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

## 6.3.3. DMM Packet Format

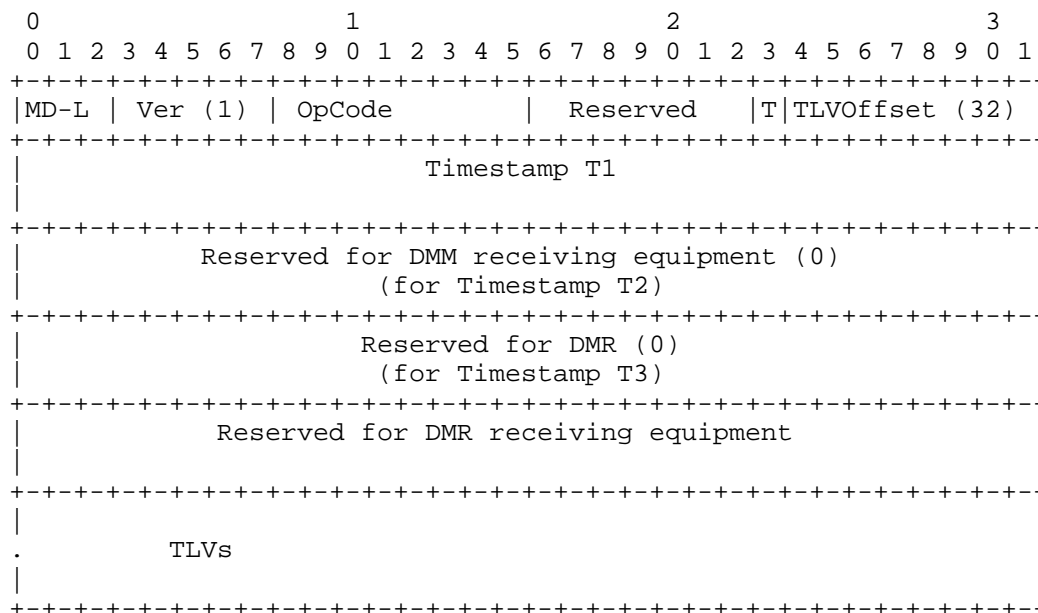


Figure 12 DMM Packet Format

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



## 6.3.4. DMR Packet Format

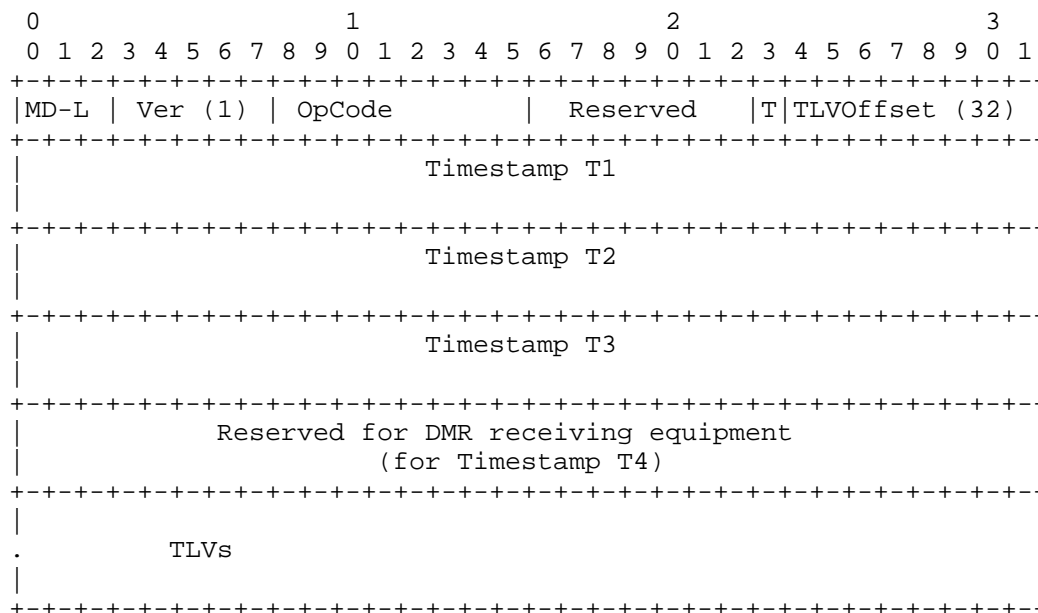


Figure 13 DMR Packet Format

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

## 7. Performance Monitoring Process

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

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

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

- o The Start Time is the time that the PM session begins.
- o The Stop Time is the time that the measurement ends.
- o The Message Period is the message transmission frequency (the time between message transmissions).
- o The Measurement Interval is the time period over which measurements are gathered and then summarized. The Measurement Interval can align with the PM Session duration, but it doesn't need to. PM messages are only transmitted during a PM Session.
- o The Repetition Time is the time between start times of the Measurement Intervals.

Figure 14 Relationship Between Different Timing Parameters

## 11. References

### 11.1. Normative References

- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFCTRILL] Perlman, R., Eastlake, D., Dutt, D., Gai, S., Ghanwani, A., "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.
- [OAM-FRAMEWK] Salam, S., Senevirathne, T., Aldrin, S., Eastlake, D., "TRILL OAM Framework", draft-ietf-trill-oam-framework (work in progress), September 2013.
- [TRILL-FM] Senevirathne, T., Finn, N., Salam, S., Kumar, D., Eastlake, D., Aldrin, S., Li, Y., "TRILL Fault Management", draft-ietf-trill-oam-fm (work in progress), July 2013.

### 11.2. Informative References

- [OAM-REQ] Senevirathne, T., Bond, D., Aldrin, S., Li, Y., Watve, R., "Requirements for Operations, Administration and Maintenance (OAM) in Transparent Interconnection of Lots of Links (TRILL)", RFC 6905, March 2013.
- [Y.1731] ITU-T Recommendation G.8013/Y.1731, "OAM Functions and Mechanisms for Ethernet-based Networks", July 2011.
- [802.1Q] "IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q(tm), 2012 Edition, October 2012.
- [IEEE1588] IEEE TC 9 Instrumentation and Measurement Society, "1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Standard, 2008.
- [MPLS-LM-DM] Frost, D., Bryant, S., "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, September 2011.
- [OAM] Andersson, L., Van Helvoort, H., Bonica, R., Romascanu, D., Mansfield, S., "Guidelines for the use of the OAM acronym in the IETF ", RFC 6291, June 2011.

## Authors' Addresses

Tal Mizrahi  
Marvell  
6 Hamada St.  
Yokneam, 20692 Israel

Email: [talmi@marvell.com](mailto:talmi@marvell.com)

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

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

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

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

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

Phone : +1 408-853-9760  
Email: [dekumar@cisco.com](mailto:dekumar@cisco.com)

Donald Eastlake 3rd  
Huawei USA R&D  
155 Beaver Street  
Milford, MA 01757 USA  
Phone: +1-508-333-2270  
Email: [d3e3e3@gmail.com](mailto:d3e3e3@gmail.com)



TRILL Working group  
Internet Draft  
Intended status: Standard Track

Tissa Senevirathne  
Norman Finn  
Samer Salam  
Deepak Kumar  
CISCO

Donald Eastlake  
Sam Aldrin  
Yizhou Li  
Huawei

July 10, 2013

Expires: January 2014

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

#### Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on January 11, 2009.

#### Copyright Notice

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

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

## Abstract

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

## Table of Contents

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



8.2. TRILL OAM Opcodes .....	25
8.3. Format of TRILL OAM TLV .....	25
8.4. TRILL OAM TLVs .....	26
8.4.1. Common TLVs between 802.lag and TRILL .....	26
8.4.2. TRILL OAM Specific TLVs .....	27
8.4.3. TRILL OAM Application Identifier TLV .....	27
8.4.4. Out Of Band Reply Address TLV .....	28
8.4.5. Diagnostics Label TLV .....	29
8.4.6. Original Data Payload TLV .....	31
8.4.7. RBridge scope TLV .....	31
8.4.8. Previous RBridge nickname TLV .....	32
8.4.9. Next Hop RBridge List TLV .....	33
8.4.10. Multicast Receiver Port count TLV .....	33
8.4.11. Flow Identifier (flow-id) TLV .....	34
8.4.12. Reflector Entropy TLV .....	35
9. Loopback Message .....	36
9.1. Loopback OAM Message format .....	36
9.2. Theory of Operation .....	36
9.2.1. Actions by Originator RBridge .....	36
9.2.2. Intermediate RBridge .....	37
9.2.3. Destination RBridge .....	37
10. Path Trace Message .....	38
10.1. Theory of Operation .....	39
10.1.1. Action by Originator RBridge .....	39
10.1.2. Intermediate RBridge .....	39
10.1.3. Destination RBridge .....	41
11. Multi-Destination Tree Verification (MTV) Message .....	41
11.1. Multi-Destination Tree Verification (MTV) OAM Message Format .....	41
11.2. Theory of Operation .....	42
11.2.1. Actions by Originator RBridge .....	42
11.2.2. Receiving RBridge .....	43
11.2.3. In scope RBridges .....	43
12. Application of Continuity Check Message (CCM) in TRILL ...	44
12.1. CCM Error Notification .....	45
12.2. Theory of Operation .....	46
12.2.1. Actions by Originator RBridge .....	46
12.2.2. Intermediate RBridge .....	47
12.2.3. Destination RBridge .....	47
13. Fragmented Reply .....	47
14. Security Considerations .....	48
15. IEEE Allocation Considerations .....	48
16. IANA Considerations .....	48
17. References .....	48
17.1. Normative References .....	48
17.2. Informative References .....	49
18. Acknowledgments .....	50

Appendix A. Backwards Compatibility .....	51
Appendix B. Base Mode for TRILL OAM .....	54
Appendix C. Unicast MAC Request .....	56

## 1. Introduction

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

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

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

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

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

## 2. Conventions used in this document

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

Acronyms used in the document include the following:

MP - Maintenance Point [TRLOAMFRM]

MEP - Maintenance End Point [TRLOAMFRM] [8021Q]

MIP - Maintenance Intermediate Point [TRLOAMFRM] [8021Q]

MA - Maintenance Association [8021Q] [TRLOAMFRM]

MD - Maintenance Domain [8021Q]

CCM - Continuity Check Message [8021Q]

LBM - Loop Back Message [8021Q]

PTM - Path Trace Message

MTV - Multi-destination Tree Verification Message

OAM - Operations, Administration, and Maintenance [RFC6291]

TRILL - Transparent Interconnection of Lots of Links [RFC6325]

FGL - Fine Grained Label [RFCfgl]

ECMP - Equal Cost Multipath

ISS - Internal Sub Layer Service [8021Q]

### 3. General Format of TRILL OAM frames

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

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

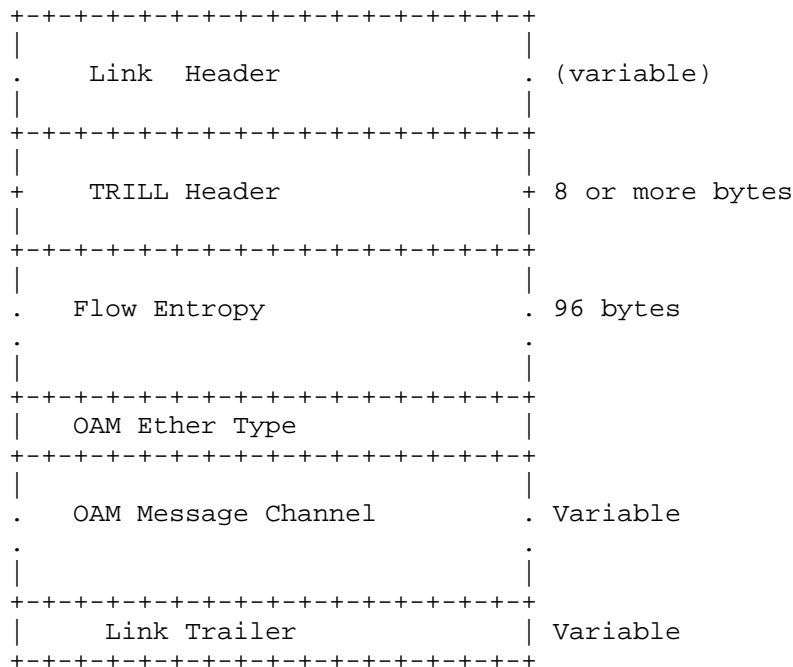


Figure 1 Format of TRILL OAM Messages

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

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

**Flow Entropy:** This is a 96-byte fixed size field. The least significant bits of the field MUST be padded with zeros, up to 96 bytes, when the flow entropy is less than 96 bytes. Flow entropy enables emulation of the forwarding behavior of the desired data packets. The Flow Entropy field starts with the Inner.MacDA. The offset of the Inner.MacDA depends on whether extensions are included or not as specified in [TRILLEXT] and [RFC6325].

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

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

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

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

### 3.1. Identification of TRILL OAM frames

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

### 3.2. Use of TRILL OAM Flag

The TRILL Header, as defined in [RFC6325], has two reserved bits. This document specifies use of the reserved bit next to Version field in the TRILL header as the Alert flag. Alert flag will be denoted by "A".

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

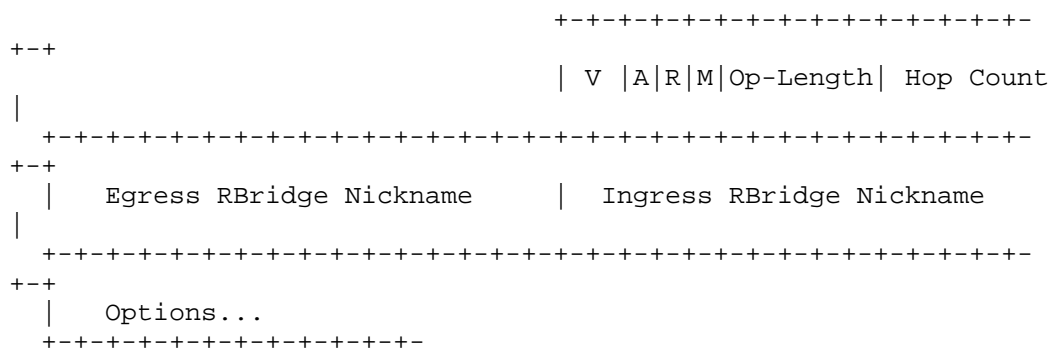


Figure 2 TRILL Header with the "A" Flag

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

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

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

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

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

### 3.3. OAM Capability Announcement

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

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

0 - OAM Capable

B - Backwards Compatible.

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

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

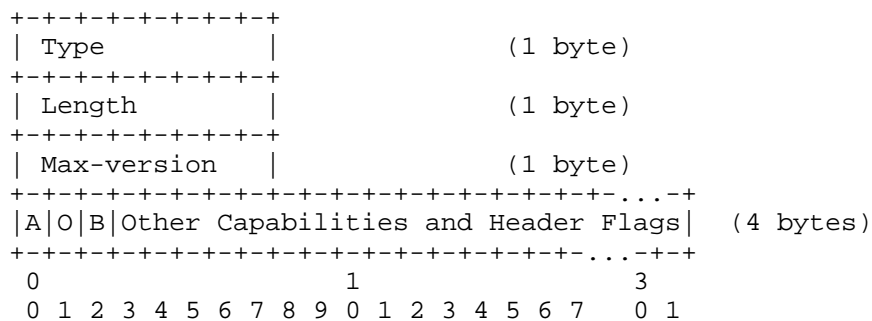


Figure 3 TRILL-VER sub-TLV [rfc6326bis] with O and B flags

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

#### 4. TRILL OAM Layering vs. IEEE Layering

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

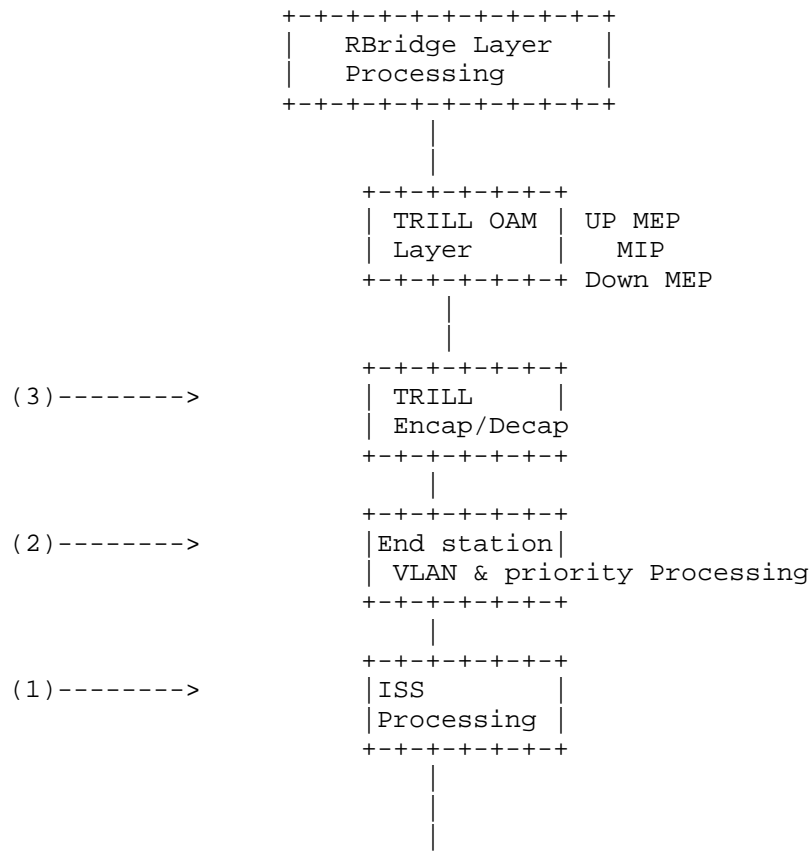


Figure 4 Placement of TRILL MP within IEEE 802.1

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

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

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



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

#### 4.1. Processing at ISS Layer

##### 4.1.1. Receive Processing

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

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

##### 4.1.2. Transmit Processing

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

#### 4.2. End Station VLAN and Priority Processing

##### 4.2.1. Receive Processing

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

##### 4.2.2. Transmit Processing

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

#### 4.3. TRILL Encapsulation and De-capsulation Layer

##### 4.3.1. Receive Processing for Unicast packets

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

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

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

#### 4.3.2. Transmit Processing for unicast packets

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

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

#### 4.3.3. Receive Processing for Multicast packets

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

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

#### 4.3.4. Transmit Processing of Multicast packets

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

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

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

#### 4.4. TRILL OAM Layer Processing

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

- o Receive indication (TRILL-HDR, iDA, iSA, iVL, M3) from RBridge layer.
- o If the TRILL Multicast Flag is set and TRILL Alert Flag is set and TRILL OAM EtherType is present then
  - o If MEP or MIP is configured on the Inner.VLAN of the packet then
    - . discard packets that have MD-LEVEL Less than that of the MEP or packets that do not have MD-LEVEL present (e.g., due to packet truncation).
    - . If MD-LEVEL matches MD-LEVEL of the MEP then
      - . Re-direct to OAM Processing (Do not forward further)
    - . If MD-LEVEL matches MD-LEVEL of MIP then
      - . Make a Copy for OAM processing and continue
  - o Else if TRILL Alert Flag is set and TRILL OAM EtherType is present then
    - o If MEP or MIP is configured on the Inner.VLAN of the packet then
      - . discard packets that have MD-LEVEL not present or MD-LEVEL is Less than that of the MEP.
      - . If MD-LEVEL matches MD-LEVEL of the MEP then
        - . Re-direct to OAM Processing (Do not forward further)
      - . If MD-LEVEL matches MD-LEVEL of MIP then
        - . Make a Copy for OAM processing and continue
    - o Else // Non OAM l Packet
      - o Continue
  - o Pass the indication (DA, SA, V, M2) to End Station VLAN processing layer.

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

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

## 5. Maintenance Associations (MA) in TRILL

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

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

For TRILL, in addition to all of the standard 802.1ag MIB definitions, each MEP's MIB contains one or more flow entropy definitions corresponding to the set of flows that the MEP monitors.

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

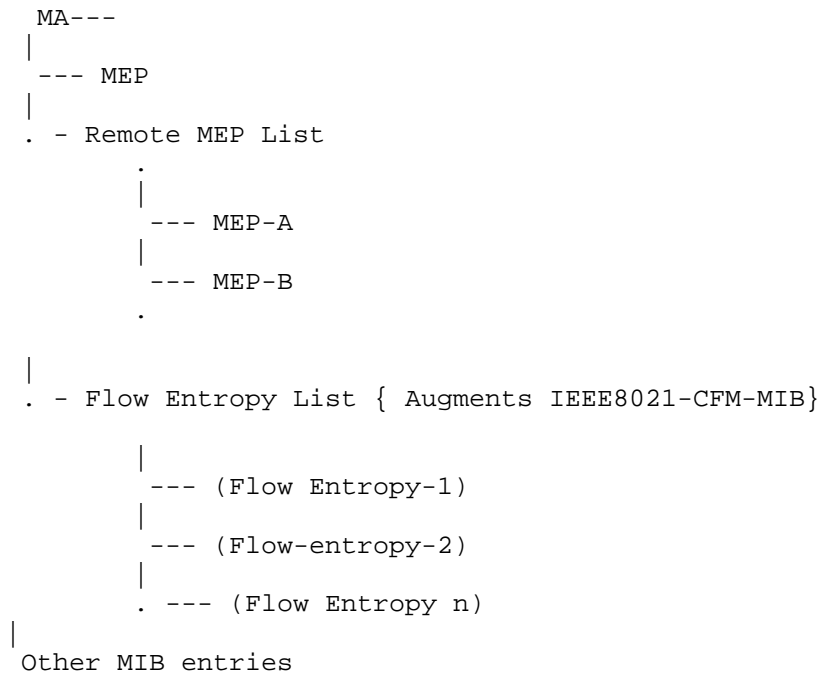


Figure 5 Correlation of TRILL augmented MIB

## 6. MEP Addressing

In IEEE 802.1ag [8021Q], OAM messages address the target MEP by utilizing a unique MAC address. In TRILL a MEP is addressed by combination of the egress RBridge nickname and the Inner VLAN/FGL.

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

1. Identify the packets that need OAM processing at the Local RBridge as specified in Section 4.

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



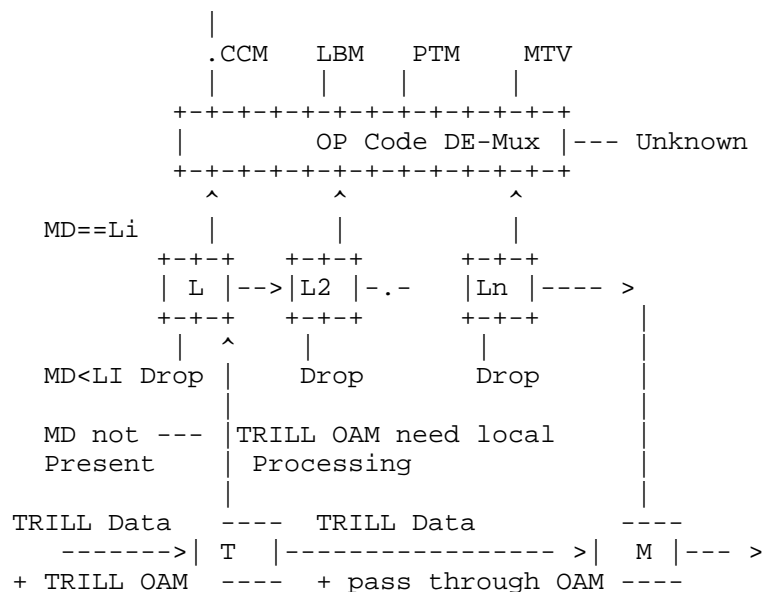


Figure 6 OAM De-Multiplexers at MEP for active SAP

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

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

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

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

### 6.1. Use of MIP in TRILL

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

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

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

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

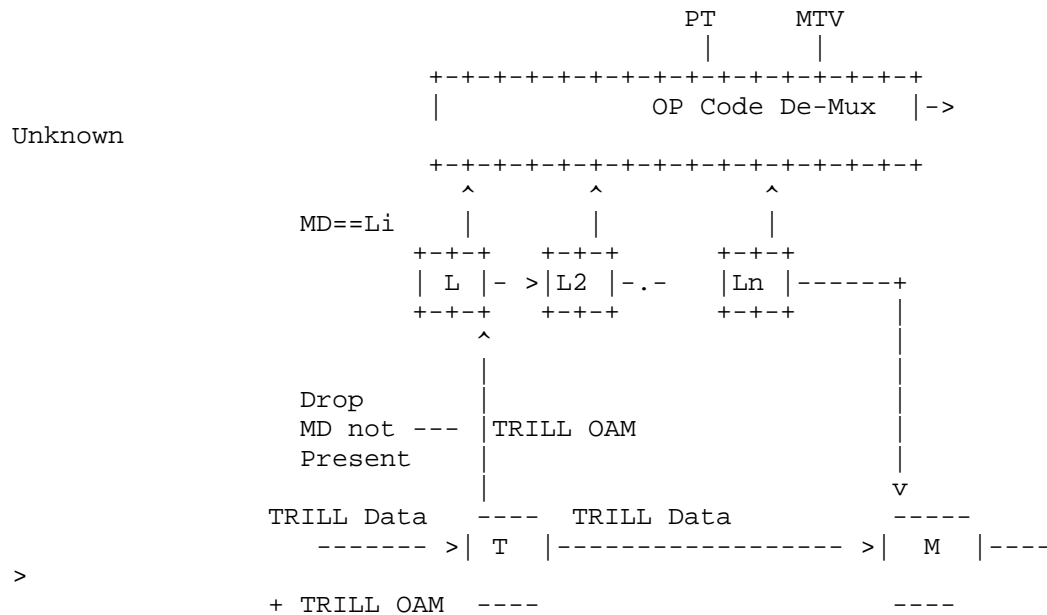


Figure 7 OAM De-Multiplexers at MIP for active SAP

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

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

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

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

Packets with unknown OpCodes are counted and discarded.

## 7. Continuity Check Message (CCM)

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

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

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

```

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

|
. - Flow Entropy List {Augments IEEE8021-CFM-MIB}
    |
    --- (Flow Entropy-1) {note we have to define
    |                       destination nickname with
    --- (Flow-entropy-2)  the flow entropy discuss}
    |
    . --- (Flow Entropy n)

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

|
:
|
- Other MIB entries

```

Figure 8 Augmentation of CCM MIB in TRILL

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

Additionally, a MEP may raise SNMP TRAPS [TRILLOAMMIB] as Alarms when a connectivity failure occurs.

## 8. TRILL OAM Message Channel

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

### 8.1. TRILL OAM Message header

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

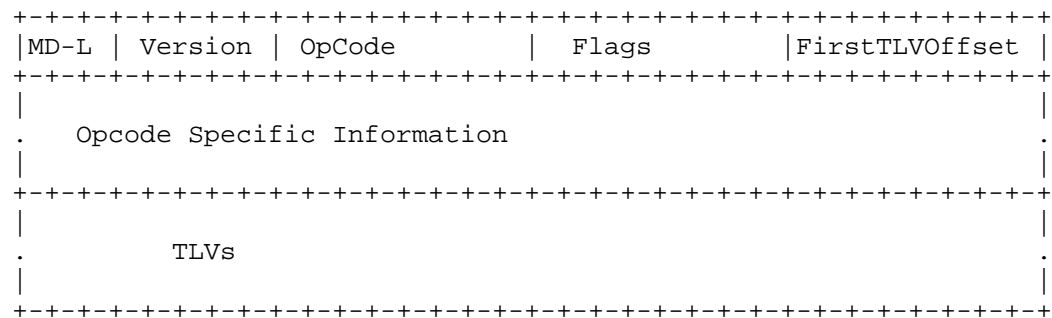


Figure 9 OAM Message Format

- o MD-L: Maintenance Domain Level (3 bits). Identifies the maintenance domain level. For TRILL, in general, this field is set to zero. However, extension of TRILL, for example to support multilevel, may create different MD-LEVELs and MD-L field must be appropriately set in those scenarios. (Please refer to [8021Q] for the definition of MD-Level)
- o Version: Indicates the version (5 bits). As specified in [8021Q]. This document does not require changing the Version defined in [8021Q].

- o **Flags:** Includes operational flags (1 byte). The definition of flags is Opcode-specific and is covered in the applicable sections.
- o **FirstTLVOffset:** Defines the location of the first TLV, in bytes, starting from the end of the FirstTLVOffset field (1 byte). (Refer to [8021Q] for the definition of the FirstTLVOffset.)

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

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

## 8.2. TRILL OAM Opcodes

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

## TRILL OAM Message Opcodes:

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

### 8.3. Format of TRILL OAM TLV

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

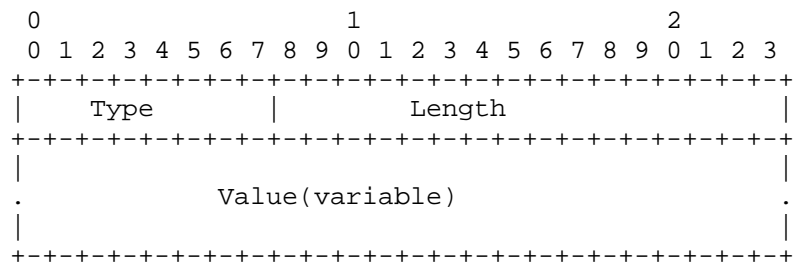


Figure 10 TRILL OAM TLV

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

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

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

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

#### 8.4. TRILL OAM TLVs

TRILL related TLVs are defined in this section. [8021Q] defined TLVs are reused, where applicable. Types 32-63 are reserved for ITU-T Y.1731. We propose to reserve Types 64-95 for TRILL OAM TLVs.

##### 8.4.1. Common TLVs between 802.1ag and TRILL

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

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



## 8.4.2. TRILL OAM Specific TLVs

As indicated above, Types 64-95 will be requested to be reserved for TRILL OAM purposes. Listed below is a summary of TRILL OAM TLVs and their corresponding codes. Format and semantics of TRILL OAM TLVs are defined in subsequent sections.

Type	TLV Name
TBD-TLV-64	TRILL OAM Application Identifier
TBD-TLV-65	Out of Band IP Address
TBD-TLV-66	Original Payload
TBD-TLV-67	Diagnostic VLAN
TBD-TLV-68	RBridge scope
TBD-TLV-69	Previous RBridge Nickname
TBD-TLV-70	TRILL Next Hop RBridge List (ECMP)
TBD-TLV-71	Multicast Receiver Availability
TBD-TLV-72	Flow Identifier
TBD-TLV-73	Reflector Entropy
TBD-TLV-74 to TBD-TLV-95	Reserved

## 8.4.3. TRILL OAM Application Identifier TLV

TRILL OAM Application Identifier TLV carries TRILL OAM application specific information. The TRILL OAM Application Identifier TLV MUST always be present and MUST be the first TLV in TRILL OAM messages. Messages that do not include the TRILL OAM Application Identifier TLV as the first TLV MUST be discarded by an RBridge, unless that RBridge is also running Ethernet CFM.

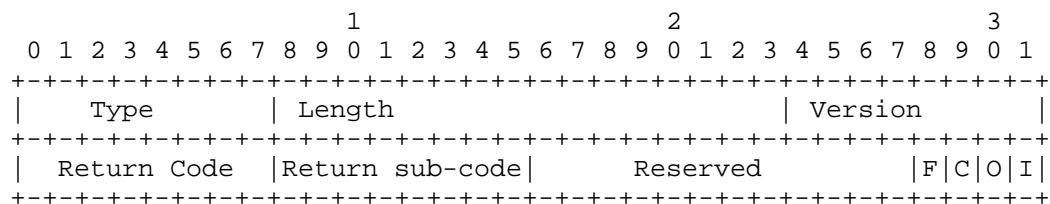


Figure 11 TRILL OAM Application Identifier TLV

Type (1 octet) = TBD-TLV-64 indicate that this is the TRILL OAM Application Identifier TLV

Length (2 octets) = 6

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

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

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

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

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

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

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

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

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

#### 8.4.4. Out Of Band Reply Address TLV

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

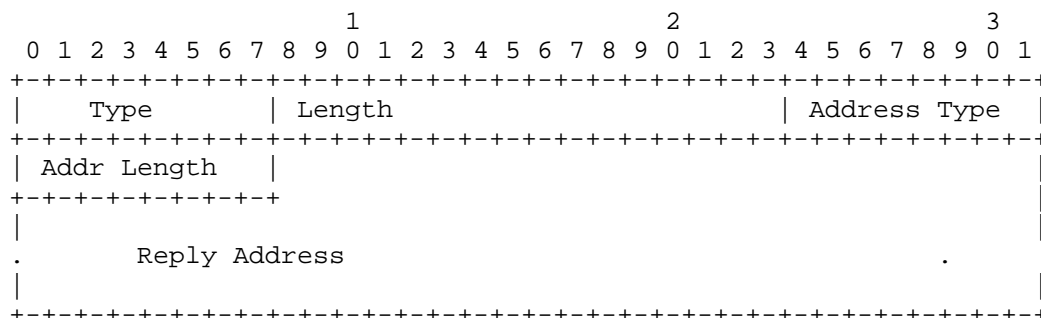


Figure 12 Out of Band IP Address TLV

Type (1 octet) = TBD-TLV-65

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

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

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

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

#### 8.4.5. Diagnostics Label TLV

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

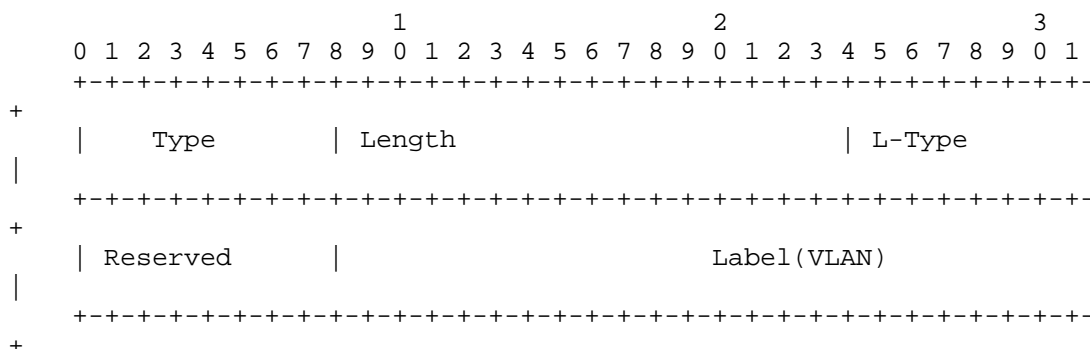


Figure 13 Diagnostic VLAN TLV

Type (1 octet) = TBD-TLV-66 indicates that this is the TRILL Diagnostic VLAN TLV

Length (2 octets) = 5

L-Type (Label type, 1 octet)

0- indicate 802.1Q 12 bit VLAN.

1 - indicate TRILL 24 bit fine grain label

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

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

## 8.4.6. Original Data Payload TLV

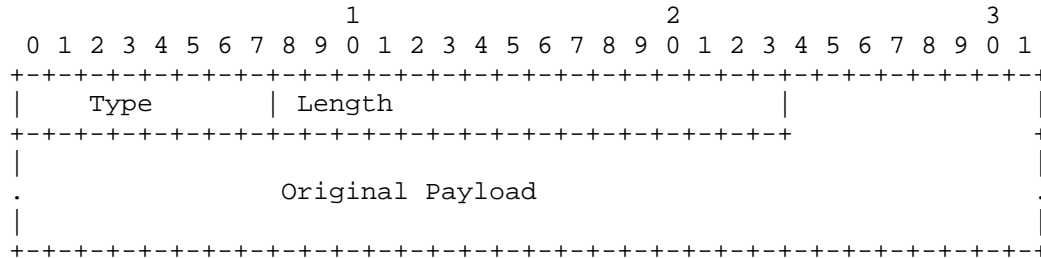


Figure 14 Original Data Payload TLV

Type (1 Octet) = TBD-TLV-67

Length (2 octets) = variable

## 8.4.7. RBridge scope TLV

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

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

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

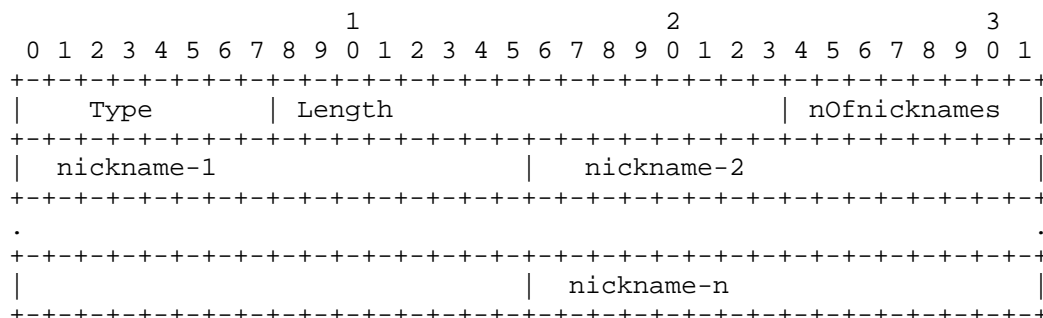


Figure 15 RBridge Scope TLV

Type (1 octet) = TBD-TLV-68 indicates that this is the "RBridge scope TLV"

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

Nickname (2 octets) = 16 bit RBridge nickname.

#### 8.4.8. Previous RBridge nickname TLV

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

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

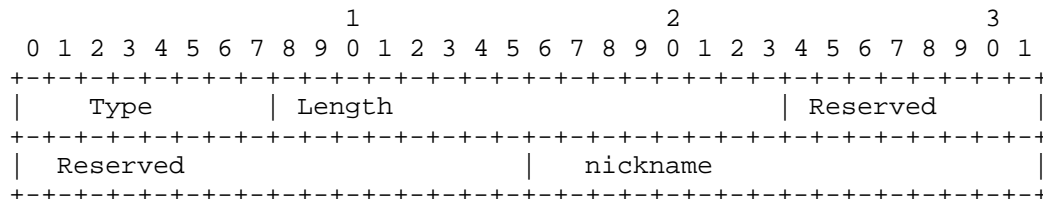


Figure 16 Previous RBridge nickname TLV

Type (1 octet) = TBD-TLV-69 indicates that this is the "Upstream RBridge nickname"

Length (2 octets) = 4.

Nickname (2 octets) = 16 bit RBridge nickname.

#### 8.4.9. Next Hop RBridge List TLV

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

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

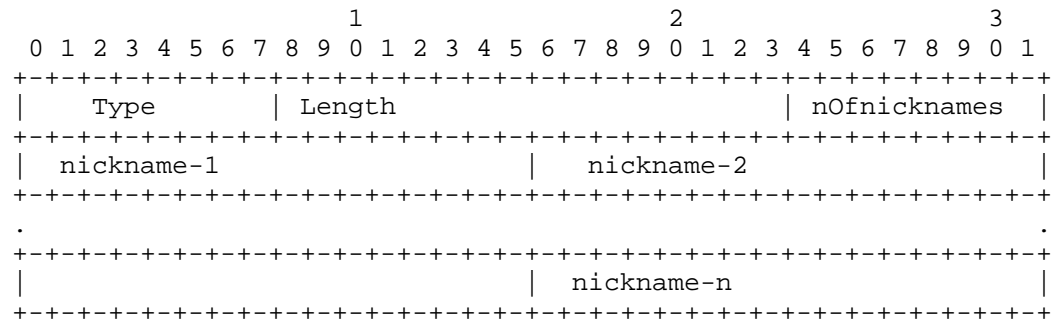


Figure 17 Next Hop RBridge List TLV

Type (1 octet) = TBD-TLV-70 indicates that this is the "Next nickname"

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

Nickname (2 octets) = 16 bit RBridge nickname.

#### 8.4.10. Multicast Receiver Port count TLV

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

Multicast Receiver Port count is an Optional TLV.

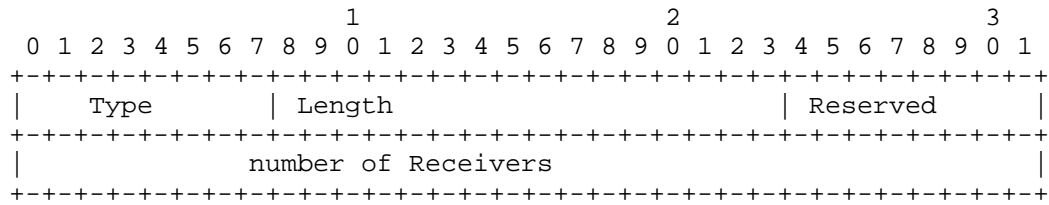


Figure 18 Multicast Receiver Availability TLV

Type (1 octet) = TBD-TLV-71 indicates that this is the "Multicast Availability TLV"

Length (2 octets) = 5.

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

#### 8.4.11. Flow Identifier (flow-id) TLV

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

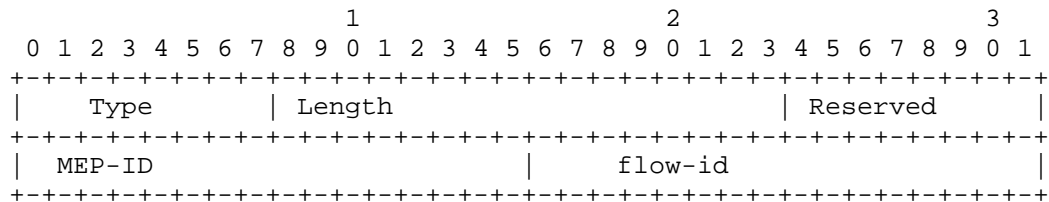


Figure 19 Flow Identifier TLV

Type (1 octet) = TBD-TLV-72

Length (2 octets) = 5.

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

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



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

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

#### 8.4.12. Reflector Entropy TLV

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

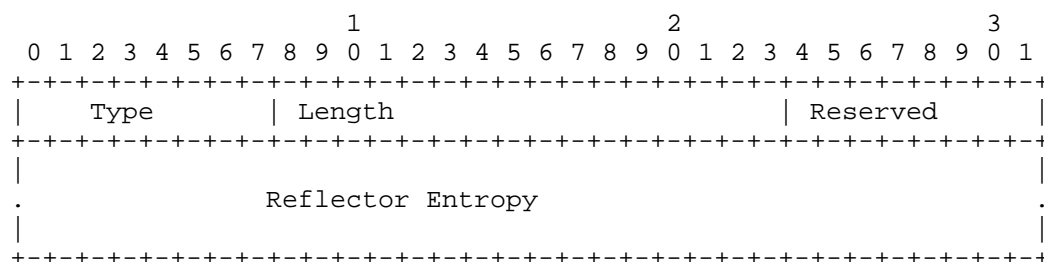


Figure 20 Reflector Entropy TLV

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

Length (1 octet) = 97.

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

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

## 9. Loopback Message

### 9.1. Loopback OAM Message format

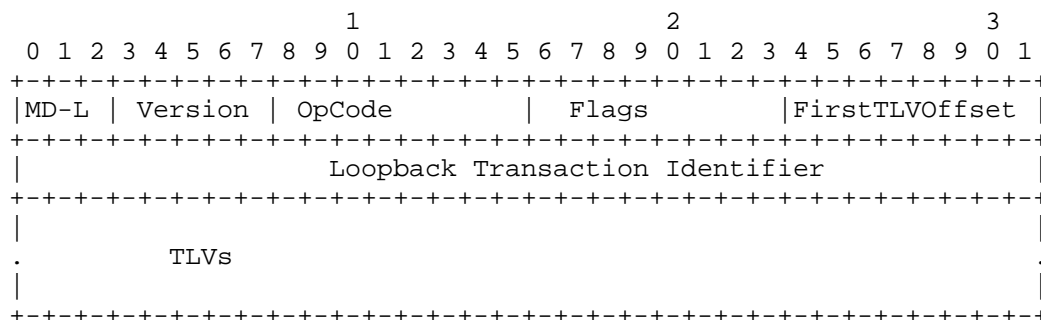


Figure 21 Loopback OAM Message Format

The above figure depicts the format of the Loopback Request and response messages as defined in [8021Q]. The Opcode for Loopback Message is set to 65 and the Opcode for the Reply Message is set to 64. The Session Identification Number is a 32-bit integer that allows the requesting RBridge to uniquely identify the corresponding session. Responding RBridges, without modification, MUST echo the received "Loopback Transaction Identifier" number..

### 9.2. Theory of Operation

#### 9.2.1. Actions by Originator RBridge

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

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

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

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

Include following OAM TLVs, where applicable

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

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

Dispatch the OAM frame for transmission.

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

#### 9.2.2. Intermediate RBridge

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

#### 9.2.3. Destination RBridge

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

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

Construction of the TRILL OAM response:

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

Set the Return Code and Return sub code to applicable values.  
Update the TRILL OAM opcode to 2 (Loopback Message Reply)

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

Include the sender ID TLV (1)

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

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

#### 10. Path Trace Message

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

[8021Q] accomplishes the objectives of the TRILL Path Trace Message using Link Trace Messages. Link Trace Messages utilize a well-known multicast MAC address. This works for [8021Q], because for 802.1 both the unicast and multicast paths are congruent. However, TRILL is multicast and unicast incongruent. Hence, TRILL OAM uses a new message format: the Path Trace message.

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

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

## 10.1. Theory of Operation

### 10.1.1. Action by Originator RBridge

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

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

Construct the TRILL OAM header: Set the opcode to Path Trace Request message type (65). Assign an applicable Session Identification number for the request. Return-code and sub-code MUST be set to zero.

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

Include following OAM TLVs, where applicable

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

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

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

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

### 10.1.2. Intermediate RBridge

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

TRILL OAM application layer further validates the received OAM frame by examining the presence of TRILL OAM Flag and OAM-

Ethertype at the end of the flow entropy and by examining the MD Level. Frames that do not contain OAM-Ethertype at the end of the flow entropy MUST be discarded.

Construction of the TRILL OAM response:

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

Set the Return Code to (2) "Time Expired" and Return sub code to zero (0). Update the TRILL OAM opcode to 64 (Path Trace Message Reply).

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

Include following TLVs

Upstream RBridge nickname TLV (69)

Reply Ingress TLV (5)

Reply Egress TLV (6)

Interface Status TLV (4)

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

Sender ID TLV (1)

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

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

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

### 10.1.1.3. Destination RBridge

Processing is identical to section 10.1.2. With the exception that TRILL OAM Opcode is set to Path Trace Reply (64).

## 11. Multi-Destination Tree Verification (MTV) Message

Multi-Destination Tree Verification messages allow verifying TRILL distribution tree integrity and pruning. TRILL VLAN/FGL and multicast pruning are described in [RFC6325] [RFC6135] and [RFC6136]. Multi-destination tree verification and Multicast group verification messages are designed to detect pruning defects. Additionally, these tools can be used for plotting a given multicast tree within the TRILL campus.

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

Implementations MAY choose to rate-limit CPU bound multicast traffic. As a result of rate-limiting or due to other congestion conditions, MTV messages may be discarded from time to time by the intermediate RBridges and the requester may be required to retransmit the request. Implementations SHOULD narrow the embedded scope of retransmission request only to RBridges that have failed to respond.

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

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

## 11.2. Theory of Operation

### 11.2.1. Actions by Originator RBridge

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

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

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

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

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

TRILL OAM Application Identifier TLV MUST be included.

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

Include RBridge scope TLV (67)

Optionally, include following TLV, where applicable

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



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

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

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

#### 11.2.2. Receiving RBridge

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

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

#### 11.2.3. In scope RBridges

Construction of the TRILL OAM response:

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

Set the Return Code to (0) and Return sub code to zero (0).  
Update the TRILL OAM opcode to 66 (Multicast Tree Verification Reply).

Include following TLVs:

Upstream RBridge nickname TLV (69)

Reply Ingress TLV (5)

Interface Status TLV (4)

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

Sender ID TLV (1)

Multicast Receiver Availability TLV (71)

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

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

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

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

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

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

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

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

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

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

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

#### 12.1. CCM Error Notification

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

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

The following example illustrates the above.

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

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

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

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

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

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

Let's Assume Flow-2 is at fault.

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

notification with MEP-A and Last good flow-id (1) and sequence number 4.

When MEP-A switches to flow-3 after transmitting flow-2, MEP-B will start receiving CCM messages. In the foregoing example it will be CCM message with Sequence Numbers 9,10,11,12,21 and so on. When in receipt of a new CCM message from a specific MEP, after a CCM timeout, the TRILL OAM will generate an SNMP Notification of CCM resume with remote MEP-ID and the first valid flow-id and the Sequence number after the CCM timeout. In the foregoing example, it is MEP-A, flow-id (1) and Sequence Number 9.

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

## 12.2. Theory of Operation

### 12.2.1. Actions by Originator RBridge

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

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

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

Include other TLVs specified in [8021Q]

Include the following optional TRILL OAM TLVs, where applicable

- o Sender ID TLV

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

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

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

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

#### 12.2.2. Intermediate RBridge

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

#### 12.2.3. Destination RBridge

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

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

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

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

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

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

### 13. Fragmented Reply

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

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

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

The TRILL OAM Application Identifier TLV MUST be included with the appropriate Final Flag field. The Final Flag, MUST, only be set on the final fragment of the reply.

#### 14. Security Considerations

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

#### 15. IEEE Allocation Considerations

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

#### 16. IANA Considerations

IANA is requested to do the following:

- Assign a multicast MAC address from the block assigned to TRILL [RFC6325]
- Set up sub-registry within the TRILL Parameters registry for block of TRILL "OAM OpCodes" (Section 8.2. )
- Set up sub-registry within the TRILL Parameters registry for TRILL "OAM TLV Types" (Section 8.4. )
- Assign a unicast MAC addressed under the IANA OUI, reserved for identification of OAM packets discussed in backward compatibility method (Appendix A) See Appendix C.

#### 17. References

##### 17.1. Normative References

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

- [RFC6325] Perlman, R., et.al., "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.
- [RFCcgl] D. Eastlake, M. Zhang, P. Agarwal, R. Perlman, D. Dutt, "TRILL: Fine-Grained Labeling", draft-ietf-trill-fine-labeling, work in progress.
- [8021Q] IEEE, "Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q-2011, August, 2011.

## 17.2. Informative References

- [RFC4379] Kompella, K. et.al, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.
- [RFC6291] Andersson, L., et.al., "Guidelines for the use of the "OAM" Acronym in the IETF" RFC 6291, June 2011.
- [RFC6361] Carlson, J. and Eastlake, D. "PPP Transparent Interconnection of Lots of Links (TRILL) Protocol Control Protocol", RFC 6361, August 2011.
- [RFC6905] Senevirathne, T. et.al "Requirements for Operations, Administration, and Maintenance (OAM) in Transparent Interconnection of Lots of Links (TRILL)", RFC 6905, March 2013.
- [RFCclcorrect] Eastlake, Donald, et.al. "TRILL: Clarifications, Corrections, and Updates, draft-ietf-trill-clear-correct, July 2012, in RFC Editor's queue.
- [TRLOAMFRM] Salam, S., et.al., "TRILL OAM Framework", draft-ietf-trill-oam-framework, Work in Progress, November, 2012.
- [TRILLEXT] Eastlake, Donald, et.al. "TRILL: Header Extension", draft-ietf-trill-rbridge-extension, June, 2012.
- [Y1731] ITU, "OAM functions and mechanisms for Ethernet based networks", ITU-T G.8013/Y.1731, July, 2011.

[Channel] D. Eastlake, et.al. , "TRILL: RBridge Channel Support", draft-ietf-trill-rbridge-channel-08.txt, in RFC Editor's queue.

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

## 18. Acknowledgments

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

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

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

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



## Appendix A. Backwards Compatibility

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

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

## A.1 Maintenance Point (MEP/MIP) Model

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

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

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

[8021Q] requires that the MEP filters or pass through OAM messages based on the MD-Level. The MD-Level is embedded deep in the OAM message. Hence, conventional methods of frame filtering may not be able to filter frames based on the MD-Level. As a result, OAM messages that must be dropped due to MD level mismatch may leak into a TRILL domain with different MD-Level.

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

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

It is also important to note that the MPLS OAM [RFC4379] framework does not include the concept of domains and OAM

filtering based on operators. It is our opinion that the lack of OAM frame filtering based on domains does not introduce significant functional deficiency or security risk.

#### A.2 Data plane encoding and frame identification

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

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

Identification methods are divided in to 4 broader groups.

Identification of Unicast non-IP OAM Flows,

Identification of Multicast non-IP OAM Flows,

Identification of Unicast IP OAM Flows and

Identification of Multicast IP OAM Flows

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

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

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

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

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

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

Figure 22 Identification of TRILL OAM Frames

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

## Appendix B.

## Base Mode for TRILL OAM

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

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

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

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

The Default MA belongs to MD-LEVEL 3.

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

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

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

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

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

Figure 23 MAID structure as defined in [8021Q]

Maintenance Domain Name Format is set to Value: 4

Maintenance Domain Name Length is set to value: 13

Maintenance Domain Name is set to: TrillBaseMode

Short MA Name Format is set to value: 3

Short MA Name Length is set to value: 2

Short MA Name is set to : FFFC

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

Please refer to [8021Q] for details.

Appendix C.

Unicast MAC Request

Applicant Name: IETF TRILL Working Group

Applicant Email: tsenevir@cisco.com

Applicant Telephone: 408-853-2291

Use Name: TRILL OAM

Document: draft-tissa-trill-oam-fm

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

identifiers: EUI-48

Size of Block requested: 1

Specify multicast, unicast, or both: Unicast

## Authors' Addresses

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

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

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

Email: nfinn@cisco.com

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

Email: ssalam@cisco.com

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

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

Donald Eastlake  
Huawei Technologies  
155 Beaver Street  
Milford, MA 01757

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

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

Email: aldrin.ietf@gmail.com

Yizhou Li  
Huawei Technologies  
101 Software Avenue,  
Nanjing 210012  
China

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





TRILL WG  
Internet-Draft  
Intended status: Standards Track  
Expires: April 4, 2014

Radia Perlman  
Intel Labs  
Fangwei Hu  
ZTE Corporation  
Donald Eastlake 3rd  
Huawei technology  
Kesava Vijaya Krupakaran  
Dell  
Ting Liao  
ZTE Corporation  
Oct 2013

TRILL Smart Endnodes  
draft-perlman-trill-smart-endnodes-02.txt

Abstract

This draft addresses the problem of the size and freshness of the endnode learning table in edge R Bridges, by allowing endnodes to volunteer for endnode learning and encapsulation/decapsulation. Such an endnode is known as a "smart endnode". Only the attached R Bridge can distinguish a "smart endnode" from a "normal endnode". The smart endnode uses the nickname of the attached R Bridge, so this solution does not consume extra nicknames.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 4, 2014.

Copyright Notice

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

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

## Table of Contents

1. Introduction . . . . .	3
2. TRILL-Hello Content . . . . .	4
2.1. Edge RBridge's TRILL-Hello . . . . .	4
2.2. Smart Endnode's TRILL-Hello . . . . .	4
3. Frame Processing . . . . .	5
3.1. Frame Processing for Smart Endnode . . . . .	5
3.2. Frame Processing for Edge RBridge . . . . .	6
4. Multi-homing . . . . .	7
5. Security Considerations . . . . .	7
6. Acknowledgements . . . . .	8
7. IANA Considerations . . . . .	8
8. Normative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

The IETF TRILL (Transparent Interconnection of Lots of Links) protocol implemented by devices called RBridges (Routing Bridges, [RFC6325]), provides optimal pair-wise data frame forwarding without configuration, safe forwarding even during periods of temporary loops, and support for multipathing of both unicast and multicast traffic. TRILL accomplishes this by using IS-IS([RFC1195]) ([RFC6165]) ([I-D.ietf-isis-rfc6326bis]) link state routing and encapsulating traffic using a header that includes a hop count. Devices that implement TRILL are called "RBridges" (Routing Bridges) or TRILL Switches.

An RBridge that attaches to endnodes is called an "edge RBridge", whereas one that exclusively forwards encapsulated frames is known as a "transit RBridge". An edge RBridge traditionally is the one that encapsulates a native Ethernet packet with a TRILL header, or that receives a TRILL-encapsulated packet and removes the TRILL header. To encapsulate, the edge RBridge must keep an "endnode table" consisting of (MAC, TRILL egress switch nickname) pairs, for those MAC addresses currently communicating with endnodes to which the edge RBridge is attached.

These table entries might be configured, received from ESADI ([I-D.ietf-trill-esadi]), looked up in a directory([I-D.ietf-trill-directory-framework]), or learned from received traffic. If the edge RBridge has many attached endnodes, this table could become large. Also, if one of the MAC addresses in the table has moved to a different switch, it might be difficult for the edge RBridge to notice this quickly, and because the edge RBridge is tunneling to the incorrect egress RBridge, the traffic will get lost.

For these reasons, it is desirable for an endnode E (whether it is a server, hypervisor, or VM) to maintain the endnode table for nodes that E is corresponding with. This eliminates the need for the attached RBridge R to know about those nodes (unless some non-smart endnode attached to R is also corresponding with those nodes), and it enables E to immediately discard an entry of (D, egress nickname), if E cannot talk to D. Then E can attempt to acquire a fresh entry for D by flooding to D, listening for ESADI, or consulting a directory.

The mechanism in this draft is that E issue a TRILL-Hello (even though E is just an endnode), indicating E's desire to act as a smart endnode, together with the set of MAC addresses that E owns, and whether E would like to receive ESADI frame. E learns from R's Hello, whether R is capable of having a smart endnode neighbor, what R's nickname is, and which trees R can use when R ingresses

multidestination frames. Although E transmits TRILL-Hellos, E does not transmit or receive LSPs.

R will accept already-encapsulated packets from E (perhaps verifying that the source MAC is indeed one of the ones that E owns, that the ingress RBridge field is R's, and if the packet is an encapsulated multidestination frame, the tree selected is one of the ones that R has claimed it will choose). When R receives (from the campus) a TRILL-encapsulated frame with R's nickname as egress, R checks whether the destination MAC address in the inner packet is one of the MAC addresses that E owns, and if so, R forwards the packet onto E's port, keeping it encapsulated.

## 2. TRILL-Hello Content

Suppose endnode E is attached to RBridge R. In order for E to act as a smart endnode, both E and R have to be signaled. The logical choice of frame to do this is TRILL-Hello.

### 2.1. Edge RBridge's TRILL-Hello

For smart endnode operation, R's TRILL-Hello must contain the following information:

- o RBridge's nickname. The nickname sub-TLV (Specified in section 2.3.2 in [I-D.ietf-isis-rfc6326bis]) could be reused here, and TLV 242 (ISIS router capability) should be updated to be carried in TRILL-Hello frame.
- o Tree roots that R can use when ingressing multidestination frames. The Tree Identifiers Sub-TLV (Specified in section 2.3.4 in [I-D.ietf-isis-rfc6326bis]) could be reused here.
- o Smart endnode neighbor list. The TRILL Neighbor TLV (Specified in section 2.5 in [I-D.ietf-isis-rfc6326bis]) could be reused.

### 2.2. Smart Endnode's TRILL-Hello

A new TLV(S-MAC TLV) is defined for smart endnode. If there are several VLANs for that smart endnode, the TLV could be filled several times in smart endnode's TRILL-Hello.

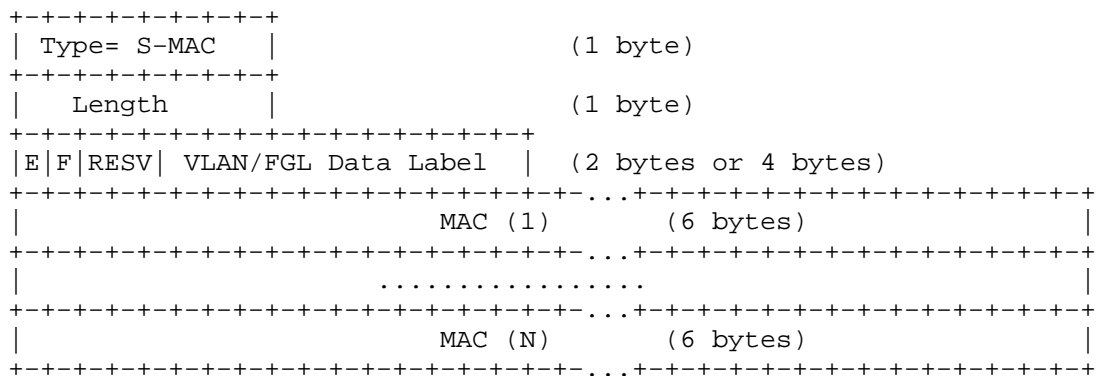


Figure 1 S-MAC TLV

- o Type: S-MAC, the value is TBD.
- o Length: Total number of bytes contained in the value field.
- o E: one bit. If it sets to 1, which indicates that the endnode could receive ESADI frame.
- o F: one bit. If it sets to 1, which indicates that the endnode supports FGL data label, otherwise, the VLAN/FGL Data Label ([I-D.ietf-trill-fine-labeling]) field is the VLAN ID.
- o RESV: 2 bits or 6 bits, is reserved for the future use. If VLAN/FGL Data Label indicates the VLAN ID(or F flag sets to 0), the RESV field is 2 bits length, otherwise it is 6 bits.
- o VLAN/FGL Data Label: This carries a 12-bits VLAN identifier or 24-bits FGL Data Label that is valid for all subsequent MAC addresses in this TLV, or the value zero if no VLAN/FGL data label is specified.
- o MAC(i): This is the 48-bit MAC address reachable from the IS that is announcing this TLV.

### 3. Frame Processing

#### 3.1. Frame Processing for Smart Endnode

Smart endnode E does not issue LSPs, nor does it receive LSPs or calculate topology. E does the following:

- o E maintains an endnode table of (MAC, nickname) of end nodes with which the smart endnode is communicating. If E is attached to multiple VLANs (or FGL), there would be a separate (MAC, nickname) table for each VLAN/FGL that E is attached to. Entries in this table are populated the same way that an edge RBridge populates the entries in its table:
  - \* learning from (source, ingress) on packets it decapsulates.
  - \* from ESADI([I-D.ietf-trill-esadi]).
  - \* by querying a directory([I-D.ietf-trill-directory-framework]).
  - \* by having some entries configured.
- o When E wishes to transmit to unicast destination D, if (D, nickname) is in E's endnode table, E encapsulates with ingress nickname=R, egress nickname as indicated in D's table entry. If D is unknown, D either queries a directory or encapsulates the packet as a multideestination frame, using one of the trees that R has specified in R's TRILL-Hello.
- o When E wishes to transmit to a multicast destination, E encapsulates the packet using one of the trees that R has specified.

The smart endnode E needs not send Hellos as frequently as normal RBridges. These hellos MAY be periodically unicast to the Appointed Forwarder R. In case R crashes and restarts, or the DRB changes, and E receives the TRILL-Hello without mentioning E, then E SHOULD send a Hello immediately. If R is AF for any of the VLANs that E claims, R MUST list E in its Hellos as a smart endnode neighbor.

### 3.2. Frame Processing for Edge RBridge

The attached RBridge R does the following:

- o If receiving an encapsulated unicast data frame from a port with a smart endnode, with R's nickname as ingress, R forwards the frame to the specified egress nickname, as with any encapsulated frame. However, R MAY filter the encapsulation frame based on the inner source MAC and VLAN (or FGL) as specified for the smart endnode. If the MAC (or VLAN/FGL) are not among the expected set of the smart endnode, the frame would be dropped by the edge RBridge.
- o If receiving an mulitdestination data TRILL frame from a port with smart endnode, RBridge R forwards the TRILL encapsulation to the TRILL campus based on the distribution tree. If there are some

normal endnodes(i.e, non-smart endnode) attached to RBridge R, R should decapsulates the frame and sends the native frame to these ports.

- o When R receives a mulicast frame from a remote RBridge, and the exit ports includes hybrid endnodes, it should send two copies of mulicast frames, one as native and the other as TRILL encapsulated frame. When smart endnode receives the encapsulated frame, it learns the remote address.

#### 4. Multi-homing

Now suppose E is attached to the TRILL campus in two places: to RBridges R1 and R2. There are two ways for this to work:

- (1) E can choose either R1 or R2's nickname, when encapsulating a frame, whether the encapsulated frame is sent via R1 or R2. If E wants to do active-active load splitting, and uses R1's nickname when forwarding through R1, and R2's nickname when forwarding through R2, which will cause the flip-floping of the endnode table entry in the remote RBridges(or smart endnodes). This issues could be solved by setting a multi-homing bit in the RESV field of the TRILL data Frame. When remote RBs or smart endnodes receive the data frame with the multi-homed bit set, the MAC entry (E, R1's nickname) and (E, R2's nickname) will be coexist as two entries for that MAC address.
- (2) R1 and R2 might indicate, in their Hello, a virtual nickname that attached end nodes may use if they are multihomed to R1 and R2, separate from R1 and R2's nicknames (which they would also list in their Hello). This would be useful if there were many end nodes multihomed to the same set of RBridges. This would be analogous to a pseudonode nickname; return traffic would go via the shortest path from the source to the endnode, whether it is R1 or R2. If E loses connectivity to R2, then E would revert to using R1's nickname. In order to avoid RPF check issue for multi-destination frame, the affinity TLV ([I-D.ietf-trill-cmt]) is recommended to be used in this solution.

#### 5. Security Considerations

For general TRILL Security Considerations, see([RFC6325]).



## 6. Acknowledgements

## 7. IANA Considerations

IANA is requested to allocate a S-MAC TLV identifier. TLV 242(ISIS router capability) is required to be updated to be carried by TRILL-Hello frame.

## 8. Normative References

[I-D.ietf-isis-rfc6326bis]

Eastlake, D., Senevirathne, T., Ghanwani, A., Dutt, D., and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", draft-ietf-isis-rfc6326bis-01 (work in progress), April 2013.

[I-D.ietf-trill-cmt]

Senevirathne, T., Pathangi, J., and J. Hudson, "Coordinated Multicast Trees (CMT) for TRILL", draft-ietf-trill-cmt-02 (work in progress), October 2013.

[I-D.ietf-trill-directory-framework]

Dunbar, L., Eastlake, D., Perlman, R., and I. Gashinsky, "TRILL (Transparent Interconnection of Lots of Links): Edge Directory Assistance Framework", draft-ietf-trill-directory-framework-07 (work in progress), August 2013.

[I-D.ietf-trill-esadi]

Zhai, H., Hu, F., Perlman, R., Eastlake, D., and O. Stokes, "TRILL (Transparent Interconnection of Lots of Links): ESADI (End Station Address Distribution Information) Protocol", draft-ietf-trill-esadi-03 (work in progress), July 2013.

[I-D.ietf-trill-fine-labeling]

Eastlake, D., Zhang, M., Agarwal, P., Perlman, R., and D. Dutt, "TRILL (Transparent Interconnection of Lots of Links): Fine-Grained Labeling", draft-ietf-trill-fine-labeling-07 (work in progress), May 2013.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

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

Requirement Levels", BCP 14, RFC 2119, March 1997.

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

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

#### Authors' Addresses

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

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

Fangwei Hu  
ZTE Corporation  
No.889 Bibo Rd  
Shanghai, 201203  
China

Phone: +86 21 68896273  
Email: hu.fangwei@zte.com.cn

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

Phone: +1-508-634-2066  
Email: d3e3e3@gmail.com

Kesava Vijaya Krupakaran  
Dell  
Olympia Technology Park  
Guindy Chennai, 600 032  
India

Phone: +91 44 4220 8496  
Email: Kesava\_Vijaya\_Krupak@Dell.com

Ting Liao  
ZTE Corporation  
No.50 Ruanjian Ave.  
Nanjing, Jiangsu 210012  
China

Phone: +86 25 88014227  
Email: liao.ting@zte.com.cn



INTERNET-DRAFT  
Intended Status: Proposed Standard  
Expires: April 24, 2014

Mingui Zhang  
Huawei  
Russ White  
Verisign  
Hongjun Zhai  
ZTE  
October 21, 2013

Control Plane Requirements for TRILL Active/Active Edge  
draft-zhang-trill-active-active-cp-req-00.txt

Abstract

TRILL Active/Active Edge enables a Multi-Chassis Link Aggregation Group to connect to multiple RBridges which can ingress and egress data traffic for the same VLAN at the same time. The purpose of introducing the TRILL Active/Active Edge is to increase the access bandwidth and resilience. This new access type puts forward new requirements for TRILL control plane. Current TRILL control plane need be extended in order to make the TRILL Active/Active Edge operational. Requirements are developed in this document as the guidelines for designing those specific control plane functions.

Status of this Memo

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

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

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

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

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

Copyright and License Notice

Copyright (c) 2013 IETF Trust and the persons identified as the

document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
2. Acronyms and Terminology . . . . .	3
2.1. Acronyms . . . . .	3
2.2. Terminology . . . . .	3
3. Control Plane Requirements . . . . .	3
3.1. Discovery . . . . .	4
3.2. Election . . . . .	4
3.3. Forwarding Information Synchronization . . . . .	4
3.4. Failure Detection and Notification . . . . .	5
3.5. Communicating Configuration Information . . . . .	5
4. Security Considerations . . . . .	5
5. IANA Considerations . . . . .	6
6. References . . . . .	6
6.1. Normative References . . . . .	6
6.2. Informative References . . . . .	6
Author's Addresses . . . . .	8

## 1. Introduction

TRILL makes use of IS-IS link state routing to provide least-cost forwarding between TRILL switches. At the edge, [RFC6349] already defines an active-standby access for end-stations. An active-active method is to be added in TRILL so that end stations are able to increase the bandwidth and resilience of their access to a TRILL campus using Multi-Chassis Link Aggregation Group (MC-LAG) [PS]. Unlike a LAN link, MC-LAG does not exchange TRILL IS-IS PDUs. The TRILL switch ports attached to the MC-LAG demarcate the edge of TRILL and no adjacency can be formed on top of the MC-LAG.

From the point of view of the end stations, the MC-LAG is treated as if it were a single link and those RBridges connected to the other end of the MC-LAG need operate as if they were a single TRILL switch. Thus an Active/Active Edge (AAE) is set up. However, it doesn't mean that RBridges in the AAE can be connected to only one MC-LAG. It's possible that one port of an RBridge is connected to one MC-LAG and the other port is connected to another.

To achieve the TRILL Active/Active Edge, some functions should be added to the current TRILL control plane. There are several possible places to host these functions. For example, the TRILL IS-IS, TRILL BFD, ESADI protocol and TRILL Channel are potential choices [BFD] [ESADI] [Channel]. This document describes the high-level requirements for these new control plane functions. When specific protocols are designed, these requirements should be followed.

## 2. Acronyms and Terminology

### 2.1. Acronyms

MC-LAG: Multi-Chassis Link Aggregation Group  
IS-IS: Intermediate System to Intermediate System  
TRILL: Transparent Interconnection of Lots of Links  
AAE: Active/Active Edge

### 2.2. Terminology

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

Familiarity with [RFC6325], [RFC6327], [6327bis] and [RFC6439] is assumed in this document.

## 3. Control Plane Requirements

This section specifies the requirements on the functions that the TRILL control plane need to provide for the AAE.

### 3.1. Discovery

When an RBridge is attached to an MC-LAG, it need to recognize other RBridges attached to this MC-LAG as in the same AAE. It also need to notify other RBridges the fact that it joins or leaves the AAE.

The identification of the AAE is REQUIRED during the discovery. The System Identifier of the MC-LAG is a choice. RBridges in an AAE MUST include this ID in the Protocol Data Units of the control plane protocol to achieve the discovery.

### 3.2. Election

RBridges per [RFC6325] run a protocol on the link to elect the Designated RBridge (DRB). The DRB performs some common tasks for the link. For example, the DRB gives the link a pseudonode nickname.

RBridges in an AAE need elect a master node to carry on common tasks for the AAE as well. Since MC-LAG disables the delivery of TRILL IS-IS PDUs. The TRILL IS-IS election protocol defined in Section 2.1 of [RFC6325] is not applicable here.

A substitute election protocol SHOULD be set up. Such protocol should reuse the decision process algorithm defined in [ISIS] to avoid introducing too much complexity into TRILL.

### 3.3. Forwarding Information Synchronization

As an example, AAE members may learn MAC addresses through data plane learning and ESADI protocol. MAC addresses learnt by one member should be shared to other members in the same AAE in order to reduce the unknown unicast traffic [PS]. As another example, the IGMP (Internet Group Management Protocol) [RFC3376] snooping on those ports attached to a MC-LAG has to be synchronized. A protocol is needed to synchronize these forwarding information among AAE members and keep the information updated in a timely manner.

MAC address may be frequently updated due to data plane learning. It is REQUIRED that the CPU is not overloaded due to the control plane MAC address update. Therefore the protocol should define a minimum updating interval.

Due to the overhead that may be produced, this protocol SHOULD be confined in the scope of the AAE members. Obviously, it SHOULD NOT be realized though the extension of TRILL IS-IS, which may otherwise



introduce a heavy burden to current TRILL's control plane. TRILL ESADI or TRILL Channel are proper candidates to realize such protocol.

#### 3.4. Failure Detection and Notification

When a link of the MC-LAG to an AAE member RBridge or this RBridge itself is failed, this failure should be detected and notified to other AAE member RBridges through the control plane as soon as possible. RBridges other than AAE member RBridges may need be notified as well so that these RBridges can change their forwarding paths to avoid the failure.

The failed AAE member RBridge leaves the AAE. It's possible that there is less than two RBridges in the AAE due to the failure. Then the AAE should be destroyed. If the AAE is not destroyed, the failure notification will trigger a re-convergence of the AAE. It is optional to establish a dedicated session such as BFD to detect the failure in order to enable a fast convergence. All AAE members MUST run into a consensus converge state just like the convergence in IGP routing protocols. The control plane protocols need take the design of the re-convergence algorithm into consideration.

#### 3.5. Communicating Configuration Information

Configuration on RBridges to enable the operation of an AAE should be minimized. Some of the configuration is local while the other is of global sense and need be conveyed through the control plane to other RBridges. An example is the Affinity Sub-TLV defined in [6326bis] and used in [CMT].

The communication of the configuration information through the control plane helps to settle mis-configuration. For example, enabled VLANs of every port attached to the same MC-LAG MUST be the same. An RBridge in an AAE can report the enabled VLANs to others through the control plane so that a mis-configured VLAN can be rectified or trigger an alarm to the management plane.

#### 4. Security Considerations

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

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

For security considerations pertain to extensions hosted by TRILL BFD and ESADI, corresponding documents should refer to the Security

Considerations in [BFD], [ESADI] and [Channel].

## 5. IANA Considerations

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

## 6. References

### 6.1. Normative References

- [RFC6325] Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.
- [RFC6327] Eastlake 3rd, D., Perlman, R., Ghanwani, A., Dutt, D., and V. Manral, "Routing Bridges (RBridges): Adjacency", RFC 6327, July 2011.
- [6327bis] D. Eastlake, R. Perlman, et al, "TRILL: Adjacency", draft-ietf-trill-rfc6327bis-01.txt, July 2013, working in progress.
- [RFC6439] Perlman, R., Eastlake, D., Li, Y., Banerjee, A., and F. Hu, "Routing Bridges (RBridges): Appointed Forwarders", RFC 6439, November 2011.
- [BFD] V. Manral, D. Eastlake, et al, "TRILL (Transparent Interconnection of Lots of Links): Bidirectional Forwarding Detection (BFD) Support", draft-ietf-trill-rbridge-bfd-07.txt, July 2012, working in progress.
- [ESADI] H. Zhai, F. Hu, et al, "TRILL (Transparent Interconnection of Lots of Links): ESADI (End Station Address Distribution Information) Protocol", draft-ietf-trill-esadi-03.txt, July 2013, working in progress.
- [6326bis] D. Eastlake, T. Senevirathne, et al, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", draft-ietf-isis-rfc6326bis-01.txt, April 2013, working in progress.
- [Channel] D. Eastlake, V Manral, et al, "TRILL: RBridge Channel Support", draft-ietf-trill-rbridge-channel-08.txt, July 2012, working in progress.

### 6.2. Informative References

- [PS] M. Zhang and D. Eastlake, "Problem Statement: TRILL Active/Active Edge", draft-zhang-trill-aggregation-04.txt, August 2013, working in progress.
- [ISIS] ISO, "Intermediate system to Intermediate system routing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002.
- [CMT] T. Senevirathne, J. Pathangi, et al, "Coordinated Multicast Trees (CMT)for TRILL", draft-ietf-trill-cmt-02.txt, November 2012, working in progress.
- [RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

Author's Addresses

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

Email: zhangmingui@huawei.com

Russ White  
Verisign  
12061 Bluemont Way  
Reston, VA 20190  
USA

Email: riwhite@verisign.com

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

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

INTERNET-DRAFT  
Intended Status: Proposed Standard  
Expires: April 24, 2014  
Updates: RFC 6325

Mingui Zhang  
Huawei  
Tissa Senevirathne  
CISCO  
Janardhanan Pathangi  
DELL  
Ayan Banerjee  
Insieme Networks  
Anoop Ghanwani  
DELL  
Donald Eastlake  
Huawei  
October 21, 2013

TRILL Resilient Distribution Trees  
draft-zhang-trill-resilient-trees-04.txt

Abstract

TRILL protocol provides layer 2 multicast data forwarding using IS-IS link state routing. Distribution trees are computed based on the link state information through Shortest Path First calculation. When a link on the distribution tree fails, a campus-wide reconvergence of this distribution tree will take place, which can be time consuming and may cause considerable disruption to the ongoing multicast service.

This document proposes to build the backup distribution tree to protect links on the primary distribution tree. Since the backup distribution tree is built up ahead of the link failure, when a link on the primary distribution tree fails, the pre-installed backup forwarding table will be utilized to deliver multicast packets without waiting for the campus-wide reconvergence, which minimizes the service disruption.

Status of this Memo

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

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

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

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

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

## Copyright and License Notice

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

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

## Table of Contents

1. Introduction . . . . .	4
1.1. Conventions used in this document . . . . .	5
1.2. Terminology . . . . .	5
2. Usage of Affinity Sub-TLV . . . . .	5
2.1. Allocating Affinity Links . . . . .	5
2.2. Distribution Tree Calculation with Affinity Links . . . . .	6
3. Resilient Distribution Trees Calculation . . . . .	7
3.1. Designating Roots for Backup Trees . . . . .	8
3.1.1. Conjugate Trees . . . . .	8
3.1.2. Explicitly Advertising Tree Roots . . . . .	8
3.2. Backup DT Calculation . . . . .	8
3.2.1. Backup DT Calculation with Affinity Links . . . . .	8
3.2.1.1. Algorithm for Choosing Affinity Links . . . . .	9
3.2.1.2. Affinity Links Advertisement . . . . .	10
3.2.2. Backup DT Calculation without Affinity Links . . . . .	10
4. Resilient Distribution Trees Installation . . . . .	10
4.1. Pruning the Backup Distribution Tree . . . . .	11
4.2. RPF Filters Preparation . . . . .	12
5. Protection Mechanisms with Resilient Distribution Trees . . . . .	12
5.1. Global 1:1 Protection . . . . .	13
5.2. Global 1+1 Protection . . . . .	13
5.2.1. Failure Detection . . . . .	14
5.2.2. Traffic Forking and Merging . . . . .	14

5.3. Local Protection . . . . .	14
5.3.1. Start Using the Backup Distribution Tree . . . . .	15
5.3.2. Duplication Suppression . . . . .	15
5.3.3. An Example to Walk Through . . . . .	15
5.4. Switching Back to the Primary Distribution Tree . . . . .	16
6. Security Considerations . . . . .	16
7. IANA Considerations . . . . .	17
Acknowledgements . . . . .	17
8. References . . . . .	17
8.1. Normative References . . . . .	17
8.2. Informative References . . . . .	18
Author's Addresses . . . . .	19

## 1. Introduction

Lots of multicast traffic is generated by interrupt latency sensitive applications, e.g., video distribution, including IP-TV, video conference and so on. Normally, a network fault will be recovered through a network wide reconvergence of the forwarding states, but this process is too slow to meet the tight Service Level Agreement (SLA) requirements on the service disruption duration. What is worse, updating multicast forwarding states may take significantly longer than unicast convergence since multicast states are updated based on control-plane signaling [mMRT].

Protection mechanisms are commonly used to reduce the service disruption caused by network faults. With backup forwarding states installed in advance, a protection mechanism is possible to restore an interrupted multicast stream in tens of milliseconds which guarantees the stringent SLA on service disruption. Several protection mechanisms for multicast traffic have been developed for IP/MPLS networks [mMRT] [MoFRR]. However, the way TRILL constructs distribution trees (DT) is different from the way multicast trees are computed under IP/MPLS, therefore a multicast protection mechanism suitable for TRILL is required.

This document proposes "Resilient Distribution Trees" (RDT) in which backup trees are installed in advance for the purpose of fast failure repair. Three types of protection mechanisms are proposed.

- o Global 1:1 protection is used to refer to the mechanism that the multicast source RBridge normally injects one multicast stream onto the primary DT. When interruption of this stream is detected, the source RBridge switches to the backup DT to inject subsequent multicast streams until the primary DT is recovered.
- o Global 1+1 protection is used to refer to the mechanism that the multicast source RBridge always injects two copies of multicast streams onto the primary DT and backup DT respectively. In the normal case, multicast receivers pick the stream sent along the primary DT and egress it to its local link. When a link failure interrupts the primary stream, the backup one will be picked until the primary DT is recovered.
- o Local protection refers to the mechanism that the RBridge attached to the failed link locally repairs the failure.

RDT may greatly reduce the service disruption caused by link failures. In the global 1:1 protection, the time cost by DT recalculation and installation can be saved. The global 1+1 protection and local protection further save the time spent on



failure propagation. A failed link can be repaired in tens of milliseconds. Although it's possible to make use of RDT to achieve load balance of multicast traffic, this document leaves that for future study.

[6326bis] defines the Affinity TLV. An "Affinity Link" can be explicitly assigned to a distribution tree or trees. This offers a way to manipulate the calculation of distribution trees. With intentional assignment of Affinity Links, a backup distribution tree can be set up to protect links on a primary distribution tree.

### 1.1. Conventions used in this document

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

### 1.2. Terminology

IS-IS: Intermediate System to Intermediate System  
TRILL: TRAnsparent Interconnection of Lots of Links  
DT: Distribution Tree  
RPF: Reverse Path Forwarding  
RDT: Resilient Distribution Tree  
SLA: Service Level Agreement  
PLR: Point of Local Repair, in this document, it is the multicast upstream RBridge connecting the failed link. It's valid only for local protection.

## 2. Usage of Affinity Sub-TLV

This document uses the Affinity Sub-TLV [6326bis] to assign a parent to an RBridge in a tree as discussed below.

### 2.1. Allocating Affinity Links

Affinity Sub-TLV explicitly assigns parents for RBridges on distribution trees. They are advertised in the Affinity Sub-TLV and recognized by each RBridge in the campus. The originating RBridge becomes the parent and the nickname contained in the Affinity Record identifies the child. This explicitly provides an "Affinity Link" on a distribution tree or trees. The "Tree-num of roots" of the Affinity Record identify the distribution trees that adopt this Affinity Link [6326bis].

Affinity Links may be configured or automatically determined using a certain algorithm [CMT]. Suppose link RB2-RB3 is chosen as an Affinity Link on the distribution tree rooted at RB1. RB2 should send

out the Affinity Sub-TLV with an Affinity Record like {Nickname=RB3, Num of Trees=1, Tree-num of roots=RB1}. In this document, RB3 does not have to be a leaf node on a distribution tree, therefore an Affinity Link can be used to identify any link on a distribution tree. This kind of assignment offers a flexibility to RBridges in distribution tree calculation: they are allowed to choose child for which they are not on the shortest paths from the root. This flexibility is leveraged to increase the reliability of distribution trees in this document.

An Affinity Sub-TLV which tries to connect two RBridges that are not adjacent MUST be ignored.

## 2.2. Distribution Tree Calculation with Affinity Links

When RBridges receive an Affinity Sub-TLV with Affinity Link which is an incoming link of RB2 (i.e., RB2 is the child on this Affinity Link), RB2's incoming links other than the Affinity Link are removed from the full graph of the campus to get a sub graph. RBridges perform Shortest Path First calculation to compute the distribution tree based on the sub graph. In this way, the Affinity Link will surely appear on the distribution tree.

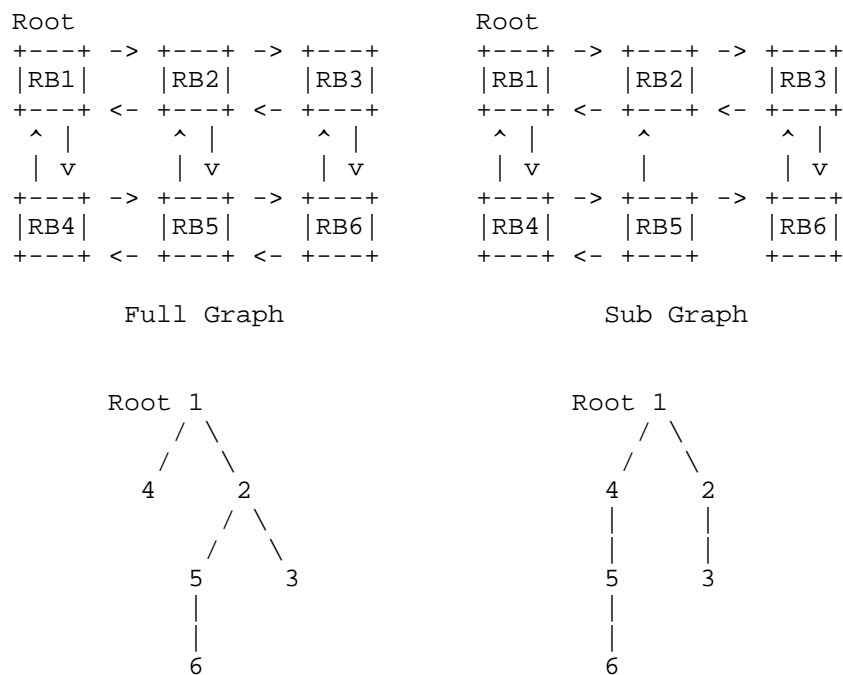


Figure 2.1: DT Calculation with the Affinity Link RB4-RB5

Take Figure 2.1 as an example. Suppose RB1 is the root and link RB4-RB5 is the Affinity Link. RB5's other incoming links RB2-RB5 and RB6-RB5 are removed from the Full Graph to get the Sub Graph. Since RB4-RB5 is the unique link to reach RB5, the Shortest Path Tree inevitably contains this link.

### 3. Resilient Distribution Trees Calculation

RBridges leverage IS-IS to detect and advertise network faults. A node or link failure will trigger a campus-wide reconvergence of distribution trees. The reconvergence generally includes the following procedures:

1. Failure detected through IS-IS control messages (HELLO) exchanging or some other method such as BFD [rbBFD];
2. IS-IS state flooding so each RBridge learns about the failure;
3. Each RBridge recalculates affected distribution trees independently;

4. RPF filters are updated according to the new distribution trees. The recomputed distribution trees are pruned per VLAN and installed into the multicast forwarding tables.

The slow reconvergence can be as long as tens of seconds, which will cause disruption to ongoing multicast traffic. In protection mechanisms, alternative paths prepared ahead of potential node or link failures are used to detour the failures upon the failure detection, therefore service disruption can be minimized.

This document will focus only on link protection. The construction of backup DT for the purpose of node protection is out the scope of this document. In order to protect a node on the primary tree, a backup tree can be setup without this node [mMRT]. When this node fails, the backup tree can be safely used to forward multicast traffic to make a detour. However, TRILL distribution trees are shared among all VLANs and Fine Grained Labels [FGL] and they have to cover all RBridge nodes in the campus [RFC6325]. A DT that does not span all RBridges in the campus may not cover all receivers of many multicast groups. (This is different from the multicast trees construction signaled by PIM [RFC4601] or mLDLP [RFC6388].)

### 3.1. Designating Roots for Backup Trees

Operators MAY manually configure the roots for the backup DTs. Nevertheless, this document aims to provide a mechanism with minimum configuration. Two options are offered as follows.

#### 3.1.1. Conjugate Trees

[RFC6325] and [ClearC] has defined how distribution tree roots are selected. When a backup DT is computed for a primary DT, its root is set to be the root of this primary DT. In order to distinguish the primary DT and the backup DT, the root RBridge MUST own multiple nicknames.

#### 3.1.2. Explicitly Advertising Tree Roots

RBridge RBl having the highest root priority nickname might explicitly advertise a list of nicknames to identify the roots of the primary and backup tree roots (See [RFC6325] Section 4.5).

### 3.2. Backup DT Calculation

#### 3.2.1. Backup DT Calculation with Affinity Links

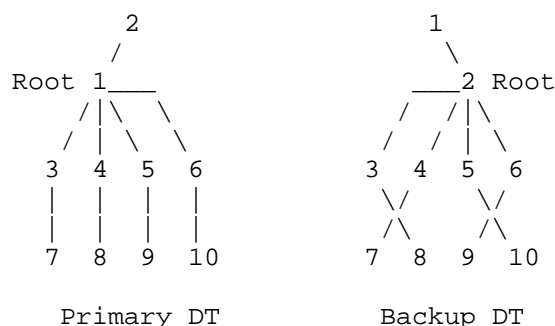


Figure 3.1: An Example of a Primary DT and its Backup DT

TRILL allows R Bridges to compute multiple distribution trees. With the intentional assignment of Affinity Links in DT calculation, this document proposes a method to construct Resilient Distribution Trees (RDT). For example, in Figure 3.1, the backup DT is set up maximally disjoint to the primary DT (The full topology is a combination of these two DTs, which is not shown in the figure.). Except for the link between RB1 and RB2, all other links on the primary DT do not overlap with links on the backup DT. It means that every link on the primary DT, except link RB1-RB2, can be protected by the backup DT.

#### 3.2.1.1. Algorithm for Choosing Affinity Links

Operators MAY configure Affinity Links to intentionally protect a specific link, such as the link connected to a gateway. But it is desirable that every R Bridge independently computes Affinity Links for a backup DT across the whole campus. This enables a distributed deployment and also minimizes configuration.

Algorithms for Maximally Redundant Trees [mMRT] may be used to figure out Affinity Links on a backup DT which is maximally disjoint to the primary DT but it only provides a subset of all possible solutions, i.e., the conjugate trees described in Section 3.1.1. In TRILL, RDT does not restrict the root of the backup DT to be the same as that of the primary DT. Two disjoint (or maximally disjointed) trees may root from different nodes, which significantly augments the solution space.

This document RECOMMENDS achieving the independent method through a slight change to the conventional DT calculation process of TRILL. Basically, after the primary DT is calculated, the R Bridge will be aware of which links will be used. When the backup DT is calculated, each R Bridge increases the metric of these links by a proper value (for safety, it's recommended to used the summation of all original link metrics in the campus but not more than  $2^{23}$ ), which gives

these links a lower priority being chosen by the backup DT by performing Shortest Path First calculation. All links on this backup DT can be assigned as Affinity Links but this is unnecessary. In order to reduce the amount of Affinity Sub-TLVs flooded across the campus, only those not picked by conventional DT calculation process ought to be recognized as Affinity Links.

#### 3.2.1.2. Affinity Links Advertisement

Similar to [CMT], every parent RBridge of an Affinity Link takes charge of announcing this link in an Affinity Sub-TLV. When this RBridge plays the role of parent RBridge for several Affinity Links, it is natural to have them advertised together in the same Affinity Sub-TLV and each Affinity Link is structured as one Affinity Record.

Affinity Links are announced in the Affinity Sub-TLV that is recognized by every RBridge. Since each RBridge computes distribution trees as the Affinity Sub-TLV requires, the backup DT will be built up consistently.

#### 3.2.2. Backup DT Calculation without Affinity Links

This section provides an alternative method to set up the disjointed backup DT.

After the primary DT is calculated, each RBridge increases the cost of those links which are already in the primary DT by a multiplier (For safety, 64x is RECOMMENDED.). It would ensure that a link appears in both trees if and only if there is no other way to reach the node (i.e. the graph would become disconnected if it were pruned of the links in the first tree.). In other words, the two trees will be maximally disjointed.

The above algorithm is similar as that defined in Section 3.2.1.1. All RBridges MUST agree on the same algorithm, then the backup DT can be calculated by each RBridge consistently and configuration is unnecessary.

### 4. Resilient Distribution Trees Installation

As specified in [RFC6325] Section 4.5.2, an ingress RBridge MUST announce the distribution trees it may choose to ingress multicast frames. Thus other RBridges in the campus can limit the amount of states which are necessary for RPF check. Also, [RFC6325] recommends that an ingress RBridge by default chooses the DT or DTs whose root or roots are least cost from the ingress RBridge. To sum up, RBridges do pre-compute all the trees that might be used so they can properly forward multi-destination packets, but only install RPF state for

some combinations of ingress and tree.

This document states that the backup DT MUST be contained in an ingress RBridge's DT announcement list and included in this ingress RBridge's LSP. In order to reduce the service disruption time, RBridges SHOULD install backup DTs in advance, which also includes the RPF filters that need to be set up for RPF Check.

Since the backup DT is intentionally built up maximally disjointed to the primary DT, when a link fails and interrupts the ongoing multicast traffic sent along the primary DT, it is probable that the backup DT is not affected. Therefore, the backup DT installed in advance can be used to deliver multicast packets immediately.

#### 4.1. Pruning the Backup Distribution Tree

The backup DT SHOULD be pruned per-VLAN. But the way a backup DT is pruned is different from the way that the primary DT is pruned. Even though a branch contains no downstream receivers, it is probable that it should not be pruned for the purpose of protection. The rule for backup DT pruning is that the backup DT should be pruned per-VLAN, eliminating branches that have no potential downstream RBridges which appear on the pruned primary DT.

It is probably that the primary DT is not optimally pruned in practice. In this case, the backup DT SHOULD be pruned presuming that the primary DT is optimally pruned. Those redundant links that ought to be pruned will not be protected.

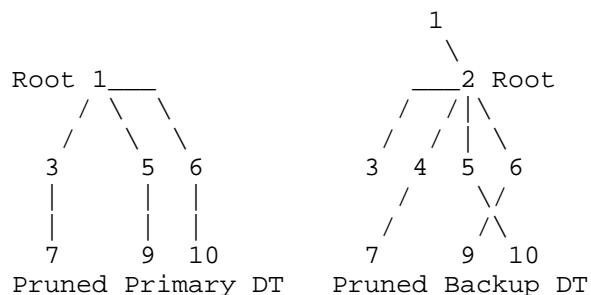


Figure 4.1: The Backup DT is Pruned Based on the Pruned Primary DT.

Suppose RB7, RB9 and RB10 constitute a multicast group MGx. The pruned primary DT and backup DT are shown in Figure 4.1. Referring back to Figure 3.1, branches RB2-RB1 and RB4-RB1 on the primary DT are pruned for the distribution of MGx traffic since there are no potential receivers on these two branches. Although branches RB1-RB2 and RB3-RB2 on the backup DT have no potential multicast receivers,

they appear on the pruned primary DT and may be used to repair link failures of the primary DT. Therefore they are not pruned from the backup DT. Branch RB8-RB3 can be safely pruned because it does not appear on the pruned primary DT.

#### 4.2. RPF Filters Preparation

RB2 includes in its LSP the information to indicate which trees RB2 might choose to ingress multicast frames [RFC6325]. When RB2 specifies the trees it might choose to ingress multicast traffic, it SHOULD include the backup DT. Other RBridges will prepare the RPF check states for both the primary DT and backup DT. When a multicast packet is sent along either the primary DT or the backup DT, it will pass the RPF Check. This works when global 1:1 protection is used. However, when global 1+1 protection or local protection is applied, traffic duplication will happen if multicast receivers accept both copies of the multicast packets from two RPF filters. In order to avoid such duplication, egress RBridge multicast receivers MUST act as merge points to activate a single RPF filter and discard the duplicated packets from the other RPF filter. In normal case, the RPF state is set up according to the primary DT. When a link fails, the RPF filter based on the backup DT should be activated.

#### 5. Protection Mechanisms with Resilient Distribution Trees

Protection mechanisms can be developed to make use of the backup DT installed in advance. But protection mechanisms already developed using PIM or mLDP for multicast of IP/MPLS networks are not applicable to TRILL due to the following fundamental differences in their distribution tree calculation.

- o The link on a TRILL distribution tree is bidirectional while the link on a distribution tree in IP/MPLS networks is unidirectional.
- o In TRILL, a multicast source node does not have to be the root of the distribution tree. It is just the opposite in IP/MPLS networks.
- o In IP/MPLS networks, distribution trees are constructed for each multicast source node as well as their backup distribution trees. In TRILL, a small number of core distribution trees are shared among multicast groups. A backup DT does not have to share the same root as the primary DT.

Therefore a TRILL specific multicast protection mechanism is needed.

Global 1:1 protection, global 1+1 protection and local protection are developed in this section. In Figure 4.1, assume RB7 is the ingress



RBridge of the multicast stream while RB9 and RB10 are the multicast receivers. Suppose link RB1-RB5 fails during the multicast forwarding. The backup DT rooted at RB2 does not include link RB1-RB5, therefore it can be used to protect this link. In global 1:1 protection, RB7 will switch the subsequent multicast traffic to this backup DT when it's notified about the link failure. In the global 1+1 protection, RB7 will inject two copies of the multicast stream and let multicast receivers RB9 and RB10 merge them. In the local protection, when link RB1-RB5 fails, RB1 will locally replicate the multicast traffic and send it on the backup DT.

### 5.1. Global 1:1 Protection

In the global 1:1 protection, the ingress RBridge of the multicast traffic is responsible for switching the failure affected traffic from the primary DT over to the backup DT. Since the backup DT has been installed in advance, the global protection need not wait for the DT recalculation and installation. When the ingress RBridge is notified about the failure, it immediately makes this switch over.

This type of protection is simple and duplication safe. However, depending on the topology of the RBridge campus, the time spent on the failure detection and propagation through the IS-IS control plane may still cause considerable service disruption.

BFD (Bidirectional Forwarding Detection) protocol can be used to reduce the failure detection time [rbBFD]. Link failures can be rapidly detected with one-hop BFD. Multi-destination BFD extends BFD mechanism to include the fast failure detection of multicast paths [mBFD]. It can be used to reduce both the failure detection and propagation time in the global protection. In multi-destination BFD, ingress RBridge need to send BFD control packets to poll each receiver, and receivers return BFD control packets to the ingress as response. If no response is received from a specific receiver for a detection time, the ingress can judge that the connectivity to this receiver is broken. In this way, multi-destination BFD detects the connectivity of a path rather than a link. The ingress RBridge will determine a minimum failed branch which contains this receiver. The ingress RBridge will switch ongoing multicast traffic based on this judgment. For example, on figure 4.1, if RB9 does not response while RB10 still responds, RB7 will presume that link RB1-RB5 and RB5-RB9 are failed. Multicast traffic will be switched to a backup DT that can protect these two links. Accurate link failure detection might help ingress RBridges to make smarter decision but it's out of the scope of this document.

### 5.2. Global 1+1 Protection

In the global 1+1 protection, the multicast source RBridge always replicates the multicast packets and sends them onto both the primary and backup DT. This may sacrifice the capacity efficiency but given there is much connection redundancy and inexpensive bandwidth in Data Center Networks, such kind of protection can be popular [MoFRR].

#### 5.2.1. Failure Detection

Egress RBridges (merge points) SHOULD realize the link failure as early as possible so that failure affected egress RBridges may update their RPF filters quickly to minimize the traffic disruption. Three options are provided as follows.

1. Egress RBridges assume a minimum known packet rate for a given data stream [MoFRR]. A failure detection timer  $T_d$  are set as the interval between two continuous packets.  $T_d$  is reinitialized each time a packet is received. If  $T_d$  expires and packets are arriving at the egress RBridge on the backup DT (within the time frame  $T_d$ ), it updates the RPF filters and starts to receive packets forwarded on the backup DT.
2. With multi-destination BFD, when a link failure happens, affected egress RBridges can detect a lack of connectivity from the ingress [mBFD]. Therefore these egress RBridges are able to update their RPF filters promptly.
3. Egress RBridges can always rely on the IS-IS control plane to learn the failure and determine whether their RPF filters should be updated.

#### 5.2.2. Traffic Forking and Merging

For the sake of protection, transit RBridges SHOULD activate both primary and backup RPF filters, therefore both copies of the multicast packets will pass through transit RBridges.

Multicast receivers (egress RBridges) MUST act as "merge points" to egress only one copy of these multicast packets. This is achieved by the activation of only a single RPF filter. In normal case, egress RBridges activate the primary RPF filter. When a link on the pruned primary DT fails, ingress RBridge cannot reach some of the receivers. When these unreachable receivers realize it, they SHOULD update their RPF filters to receive packets sent on the backup DT.

#### 5.3. Local Protection

In the local protection, the Point of Local Repair (PLR) happens at the upstream RBridge connecting the failed link. It is this RBridge

that makes the decision to replicate the multicast traffic to recover this link failure. Local protection can further save the time spent on failure notification through the flooding of LSPs across the campus. In addition, the failure detection can be speeded up using [rbBFD], therefore local protection can minimize the service disruption within 50 milliseconds.

Since the ingress RBridge is not necessarily the root of the distribution tree in TRILL, a multicast downstream point may not be the descendants of the ingress point on the distribution tree. Moreover, distribution trees in TRILL are bidirectional and do not share the same root. There are fundamental differences between the distribution tree calculation of TRILL and those used in PIM and mLDP, therefore local protection mechanisms used for PIM and mLDP, such as [mMRT] and [MoFRR], are not applicable here.

#### 5.3.1. Start Using the Backup Distribution Tree

The egress nickname TRILL header field of the replicated multicast TRILL data packets specifies the tree on which they are being distributed. This field will be rewritten to the backup DT's root nickname by the PLR. But the ingress of the multicast frame MUST remain unchanged. This is a halfway change of the DT for multicast packets. Afterwards, the PLR begins to forward multicast traffic along the backup DT. This is a change from [RFC6325] which specifies that the egress nickname in the TRILL header of a multi-destination TRILL data packet must not be changed by transit RBridges.

In the above example, if PLR RB1 decides to send replicated multicast packets according to the backup DT, it will send it to the next hop RB2. .

#### 5.3.2. Duplication Suppression

When a PLR starts to send replicated multicast packets on the backup DT, some multicast packets are still being sent along the primary DT. Some egress RBridges might receive duplicated multicast packets. The traffic forking and merging method in the global 1+1 protection can be adopted to suppress the duplication.

#### 5.3.3. An Example to Walk Through

The example used in the above local protection is put together to get a whole "walk through" below.

In the normal case, multicast frames ingressed by RB7 with pruned distribution on primary DT rooted at RB1 are being received by RB9 and RB10. When the link RB1-RB5 fails, the PLR RB1 begins to

replicate and forward subsequent multicast packets using the pruned backup DT rooted at RB2. When RB2 gets the multicast packets from the link RB1-RB2, it accepts them since the RPF filter {DT=RB2, ingress=RB7, receiving links=RB1-RB2, RB3-RB2, RB4-RB2, RB5-RB2 and RB6-RB2} is installed on RB2. RB2 forwards the replicated multicast packets to its neighbors except RB1. When the multicast packets reach RB6 where both RPF filters {DT=RB1, ingress=RB7, receiving link=RB1-RB6} and {DT=RB2, ingress=RB7, receiving links=RB2-RB6 and RB9-RB6} are active. RB6 will let both multicast streams through. Multicast packets will finally reach RB9 where the RPF filter is updated from {DT=RB1, ingress=RB7, receiving link=RB5-RB9} to {DT=RB2, ingress=RB7, receiving link=RB6-RB9}. RB9 will egress the multicast packets on to the local link.

#### 5.4. Switching Back to the Primary Distribution Tree

Assume an RBridge receives the LSP that indicates a link failure. This RBridge starts to calculate the new primary DT based on the topology with the failed link. Suppose the new primary DT is installed at  $t_1$ .

The propagation of LSPs around the campus takes time. For safety, we assume all RBridges in the campus have converged to the new primary DT at  $t_1 + T_s$ . By default,  $T_s$  (the "settling time") is set to 30s but is configurable. At  $t_1 + T_s$ , the ingress RBridge switches the traffic from the backup DT back to the new primary DT.

After another  $T_s$  (at  $t_1 + 2T_s$ ), no multicast packets are being forwarded along the old primary DT. The backup DT should be updated according to the new primary DT. The process of this update under different protection types are discussed as follows.

- a) For the global 1:1 protection, the backup DT is simply updated at  $t_1 + 2T_s$ .
- b) For the global 1+1 protection, the ingress RBridge stops replicating the multicast packets onto the old backup DT at  $t_1 + T_s$ . The backup DT is updated at  $t_1 + 2T_s$ . It MUST wait for another  $T_s$ , during which time period all RBridges converge to the new backup DT. At  $t_1 + 3T_s$ , the ingress RBridge MAY start to replicate multicast packets onto the new backup DT.
- c) For the local protection, the PLR stops replicating and sending packets on the old backup DT at  $t_1 + T_s$ . It is safe for RBridges to start updating the backup DT at  $t_1 + 2T_s$ .

#### 6. Security Considerations

This document raises no new security issues for TRILL.

For general TRILL Security Considerations, see [RFC6325].

## 7. IANA Considerations

No new registry or registry entries are requested to be assigned by IANA. The Affinity Sub-TLV has already been defined in [6326bis]. This document does not change its definition. RFC Editor: please remove this section before publication.

## Acknowledgements

The careful review from Gayle Noble is gracefully acknowledged. The authors would like to thank the comments and suggestions from Erik Nordmark, Donald Eastlake, Fangwei Hu, Hongjun Zhai and Xudong Zhang.

## 8. References

### 8.1. Normative References

- [6326bis] D. Eastlake, T. Senevirathne, et al., "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", draft-ietf-isis-rfc6326bis-01.txt, work in Progress.
- [CMT] T. Senevirathne, J. Pathangi, et al, "Coordinated Multicast Trees (CMT) for TRILL", draft-ietf-trill-cmt-02.txt, work in progress.
- [RFC6325] R. Perlman, D. Eastlake, et al, "Rbridges: Base Protocol Specification", RFC 6325, July 2011.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.
- [RFC6388] Wijnands, IJ., Minei, I., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, November 2011.
- [rbBFD] V. Manral, D. Eastlake, et al, "TRILL (Transparent Interconnection of Lots of Links): Bidirectional Forwarding Detection (BFD) Support", draft-ietf-trill-rbridge-bfd-07.txt, work in progress.
- [ClearC] Eastlake, D., M. Zhang, A. Ghanwani, V. Manral, A.

Banerjee, "TRILL: Clarifications, Corrections, and Updates" draft-ietf-trill-clear-correct, in RFC Editor's queue.

## 8.2. Informative References

- [mMRT] A. Atlas, R. Kebler, et al., "An Architecture for Multicast Protection Using Maximally Redundant Trees", draft-atlas-rtgwg-mrt-mc-arch-02.txt, work in progress.
- [MoFRR] A. Karan, C. Filsfils, et al., "Multicast only Fast Re-Route", draft-ietf-rtgwg-mofrr-02.txt, work in progress.
- [mBFD] D. Katz, D. Ward, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-02.txt, work in progress.
- [FGL] D. Eastlake, M. Zhang, P. Agarwal, R. Perlman, D. Dutt, "TRILL (Transparent Interconnection of Lots of Links): Fine-Grained Labeling", draft-ietf-trill-fine-labeling, in RFC Editor's queue.

## Author's Addresses

Mingui Zhang  
Huawei Technologies Co.,Ltd  
Huawei Building, No.156 Beiqing Rd.  
Beijing 100095 P.R. China

Email: zhangmingui@huawei.com

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

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

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

Phone: +91 44 4220 8400  
Email: Pathangi\_Janardhanan@Dell.com

Ayan Banerjee  
Insieme Networks  
210 W Tasman Dr,  
San Jose, CA 95134

Email: ayabaner@gmail.com

Anoop Ghanwani  
Dell  
350 Holger Way  
San Jose, CA 95134

Phone: +1-408-571-3500  
Email: Anoop@alumni.duke.edu

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

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