TRILL Working Group D. Bond
Internet-Draft UNH-IOL
Intended status: Standards Track V. Manral
Expires: September 12, 2011 IP Infusion Inc.
March 11, 2011

RBridges: Operations, Administration, and Maintenance (OAM) Support
draft-bond-trill-rbridge-oam-01

Abstract

The IETF has standardized RBridges, devices that implement the TRILL protocol, a solution for transparent shortest-path frame routing in multi-hop networks with arbitrary topologies, using a link-state routing protocol technology and encapsulation with a hop-count. As RBridges are deployed in real-world situations, operators will need tools for debugging problems that arise. This document specifies a set of RBridge features for operations, administration, and maintenance purposes in RBridge campuses. The features specified in this document include tools for traceroute, ping, and error reporting.

Table of Contents

1. Introduction

The IETF has standardized RBridges, devices that implement the TRILL protocol, a solution for transparent shortest-path frame routing in multi-hop networks with arbitrary topologies, using a link-state routing protocol technology and encapsulation with a hop-count (RFCtrill [I-D.ietf-trill-rbridge-protocol]). As RBridges are deployed, operators will face problems that require tools for troubleshooting of connectivity issues in the network. TRILL uses IS-IS for the control plane. IS-IS has a link-state database which contains the information of all links in the TRILL domain and IS-IS has a routing table. This information can be used for trouble shooting purposes. Simply being able to view the link-state database and routing table is insufficient for the requirements of operations, administration, and maintenance (OAM).

In addition, RBridges should support SNMP, as described in RFCtrill [I-D.ietf-trill-rbridge-protocol] and RBridgeMIB [I-D.ietf-trill-rbridge-mib]. SNMP, the routing table, and the link-state database are insufficient as the only OAM tools because while the control plane within an RBridge campus may be functioning successfully the data plane may not be. This motivates the need for OAM tools that allow an operator to test the data plane. Protocols such as IP, MPLS, and IEEE 802.1 have features enabling an operator to exercise the data plane (RFC 4443 [RFC4443], RFC 0792 [RFC0792], IEEE 802.1ag [IEEE.802-1ag]). There is a need for a similar set of tools in TRILL.

Likewise, there is a need for error reporting capabilities inside an RBridge campus. For instance, if a TRILL Inner.VLAN tag has an illegal value there should be a way for devices to report this error. This would allow administrators of an RBridge campus to quickly locate a problem device in the network. This document specifies a set of RBridge features for operations, administration, and maintenance purposes in RBridge campuses along with a frame format. The features specified in this document include tools for traceroute, ping, and error reporting. Section 3 of this document specifies the general usage of a defined message format. Section 4 specifies some additional applications of the message format. Section 5 specifies the format of the messages on the wire.

1.1. Requirements Language

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

2. Acronyms

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

- o RBridge - Routing Bridge
- o SA - Source Address
- o SNMP - Simple Network Management Protocol
- o TLV - Type, Length, Value
- o TRILL - TRansparent Interconnection of Lots of Links
- o VLAN - Virtual Local Area Network

3. TRILL OAM Message

To facilitate message passing as needed by the OAM requirements, the TRILL OAM Channel ([I-D.eastlake-trill-rbridge-channel]) is utilized. The TRILL Header extended flag MAY be set if so desired.

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

The specification of rate limiting is beyond the scope of this document. An RBridge SHOULD maintain counters for each type of error generated.

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

```
+----------------------------+
|     Outer Link Header      |
+----------------------------+
|        TRILL Header        |
+----------------------------+
|     Inner Link Header      |
+----------------------------+
|  TRILL OAM Channel Header  |
+----------------------------+
| OAM Protocol Spec. Payload |
+----------------------------+
```

Application Frame

Figure 1

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

Error Notification Frame

Figure 2

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

Frame Type	Field	Value
Application or Error	Inner.MacSA	If the Inner.MacDA of the received frame is one of the MAC addresses of the RBridge generating the frame, the value MUST be that MAC address. Otherwise, it MUST be one of the RBridge’s MAC addresses.
Application or Error	Inner.MacDA	The value SHOULD be All-OAM-RBridges . The Inner.MacDA MAY be other values as specified in subsequent sections.
Application or Error	Inner.VLAN ID	The value MUST be one of the VLANs the egress RBridge advertises connectivity on. If the frame is generated in response to another frame it MUST be copied from the received frame.
Application or Error	Ingress RBridge nickname	If the egress RBridge nickname of the received frame is a nickname of the RBridge generating the frame, then the value MUST be that nickname. Otherwise, it MUST be one of the RBridge’s nicknames.
Application or Error	Egress RBridge nickname	The value MUST be the ingress RBridge nickname of the received frame. If the ingress RBridge nickname received is unknown or reserved the frame MUST be generated on the port the frame was received on with an Outer.MacDA and egress RBridge nickname of the RBridge that transmitted the invalid frame.

Error	Offending Encapsulated Frame	The value MUST be N bytes of the frame which had the error where N is the minimum of the frame size and the number of bytes that would bring the resulting error frame up to 1470 bytes. This MUST include the TRILL header and MUST NOT include the link-layer header.
Error	M Bit	The value MUST be zero.
Application or Error	Inner.Priority	The value SHOULD be one less than the priority of the received frame, but not less than the lowest priority. Defaults to zero for sent frames.

Table 1: Frame Field Values

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

4. RBridge Tools

This section specifies a number of RBridge OAM tools. For classification purposes they are divided into two sections, applications and error tools.

4.1. Application RBridge Tools

4.1.1. Hop Count Traceroute

The ability to trace the path the data takes through the network is an invaluable debugging tool. RBridge traceroute provides this functionality through use of the TRILL OAM message (See Section 3). In a hop-count traceroute, the originating RBridge starts by transmitting one TRILL data frame with a TRILL OAM message. This message contains a protocol code of an echo request. (See Section 5.2.1.1) The ingress RBridge MUST be the RBridge originating the frame.

When a traceroute is initiated, it is either targeting a known unicast target or a multi-destination target as specified by the operator. If the hop-count traceroute is for a known unicast target, the egress RBridge is the destination RBridge to which connectivity will be checked and the M bit MUST be zero. Otherwise, if the hop-count traceroute is for a multi-destination target, the egress RBridge is the distribution tree nickname for the traceroute. Multi-destination targets are handled the same as known unicast targets but require a small amount of additional logic as specified in Section 4.1.1.1.

The first echo request frame transmitted MUST have a hop-count of one. The RBridge will continue transmitting these echo requests, incrementing the hop-count by one each time until a hop-count error notification is received from the destination. Each of these requests in turn will generate a hop-count error notification until the egress RBridge is reached. If a transit RBridge decrements the hop-count by more than one it may transmit multiple hop-count error notifications.

The purpose of the traceroute is to confirm connectivity of the data plane, and therefore options such as a flow ID or a security option MAY be included. If an RBridge supports equal-cost multi-pathing (ECMP) or load balancing, the RBridge SHOULD allow operators to specify which flow the traceroute is assigned to. There is no need for all RBridges to use the same assignment method. Being able to specify the flow allows operators to test the path taken by data through the data plane. The purpose of the frame is to mimic a data frame that follows the same path through the data plane that a ’real’ data frame would.

The echo request MAY have an arbitrary 32-bit unsigned integer sequence number to assist in matching reply messages to the request. This is important for the hop-count traceroute since replies may return to the ingress RBridge in a different order then their matching requests were sent.

The Inner.VLAN, Inner.MacSA, and Inner.MacDA SHOULD default to the values specified in Table 1. RBridges SHOULD provide an option to change these values to assign the TRILL data frame to a flow.

The replying RBridge MUST include its 16-bit port ID from the port on which the hop-count error generating frame was received in the incoming port field of the reply. It MUST also include its 16-bit port ID from the port on which the frame would be forwarded if the frame did not have a hop-count error. A port ID of 0xFFFF indicates the frame was consumed by the RBridge itself. Finally the reply MUST include the 16-bit nickname of the next hop RBridge the frame would have been sent to if there were no error. If the request is a multi-destination frame, this field MUST be set to the nickname of the RBridge the frame was received from. This is the previous hop RBridge. This is to facilitate knowledge of a more precise path through the campus as seen in RFC 5837 [RFC5837].

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

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

4.1.1.1. Multi-Destination Targets

For multi-destination targets at each branch in the tree the tagged frame will be replicated causing each RBridge in the tree, possibly pruned by VLAN and/or multicast group, to send a response to the echo request. If all RBridges in the possibly pruned distribution tree support the echo request message, then the ingressing RBridge will receive an echo reply from each of them. This is in contrast to a known unicast tagged frame where only the RBridges along the path from ingress to egress transmit the error notification. The ingressing RBridge can compile all of these replies, using the parent pointers located in the nexthop nickname field, into an output of the tree the traffic traversed. In the case that a non-valid distribution tree nickname is specified the traceroute frames SHOULD still be generated. The traceroute application MUST report any errors received, such as an invalid distribution tree nickname, caused by the hop-count traceroute frames. RBridges receiving a multicast destination echo request MUST NOT transmit an echo reply if the multi-destination bit is set. Echo requests that are not used with the hop-count traceroute come from the ping tool, and pings are not valid to multi-destination traffic. In a hop-count traceroute devices will already be transmitting a hop-count error notification and so there is no reason to transmit a double set of replies. A multi-destination hop-count traceroute does not stop when an echo reply is received. It stops when the transmitted hopcount reaches 0x3F.

4.1.1.2. Hop Count Traceroute Example

Figure 3 contains a campus with three RBridges. Consider a hop-count traceroute from RB0 to RB2.

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

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

Hop Count Traceroute Example Topology

Figure 3

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

Below are some select fields for the frames:

Frame #	Ingress RBridge	Egress RBridge	TRILL OAM Protocol	Sequence Number	Hop Count
(1)	RB0	RB2	Echo Request	1	1
(2)	RB1	RB0	Hop Count Error	1	N/A
(3) @ RB1	RB0	RB2	Echo Request	2	2
(3) @ RB2	RB0	RB2	Echo Request	2	1
(4) @ RB1	RB2	RB0	Hop Count Error	2	N/A

(4) @ RB0	RB2	RB0	Hop Count Error	2	N/A

Table 2: Hop Count Traceroute Example Frames

For example, if the nicknames for RB0, RB1, and RB2 are 0x0001, 0x0002, and 0x0003 respectively, the console output from such a trace might be:

Hop Count Tracing

RBridge	Incoming Port Id	Outgoing Port Id	RBridge Nexthop Nickname
0x0001	0xFFFF	0x0001	0x0002
0x0002	0x0000	0x0001	0x0003
0x0003	0x0000	0xFFFF	0x0000

Table 3: Hop Count Traceroute Example Output

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

4.1.2. RBridge Ping

Ping is a tool for verifying RBridge connectivity. As with an RBridge traceroute, the ping-originating RBridge transmits one or more TRILL data frames with a TRILL OAM message. This message contains the code of an echo request (See Section 5.2.1.1). The ingress RBridge MUST be the RBridge-originating frame. The egress RBridge is the destination RBridge to which connectivity will be checked. The M bit MUST be zero.

As with RBridge traceroute, options such as a flow ID or a security option MAY be included. If an RBridge supports equal-cost multi-pathing (ECMP) or load balancing, the RBridge SHOULD allow operators to specify which flow the ping is assigned to. There is no need for all RBridges to use the same assignment method. This ping traffic, once again, will mimic real traffic through the network, like traceroute traffic as previously specified in Section 4.1.1.

The echo request MAY have an arbitrary 32-bit unsigned integer sequence number to assist in matching reply messages to the request. In most circumstances, a single echo request is needed to complete the ping but it might be desirable for a single RBridge to ping multiple egress RBridges, or trace differing flows simultaneously. Assigning differing sequence numbers to each frame aids in matching which trace the reply belongs to.

The Inner.VLAN, Inner.MacSA, and Inner.MacDA SHOULD default to the values specified in Table 1. RBridges SHOULD provide the ability to change these values as to assign the TRILL data frame to a flow. The payload of the frame is arbitrary and MAY contain any value. This value can have an influence on which flow the frame is assigned to.

RBridges implementing ping MAY issue a reply in response to this request. See Section 8 for reasons on some RBridges are allowed to choose not to respond to a request. If an RBridge chooses to respond to the request, the reply MUST consist of one TRILL data frame per request with an OAM message containing the protocol code of an echo reply. The echo reply MUST have the same sequence number as the request being matched.

For the echo reply the ingress RBridge field MUST be the reply-originating RBridge’s nickname. The egress RBridge MUST be the request-originating RBridge’s nickname. The Inner.VLAN, Inner.MacSA, and Inner.MacDA SHOULD default to the values specified in Table 1. The Outer.VLAN ID MUST be preserved. The M bit MUST be zero.

The reply-originating RBridge MUST include its 16-bit port ID from the port on which the request was received in the incoming port field of the reply. It MUST also include its 16-bit port ID from the port on which the frame is forwarded. A port ID of 0xFFFF indicates the frame was consumed by the RBridge itself. The nickname field in the generated frame MUST be set to all zeros on transmission and ignored on reception.

The Internal Hop Count field of the reply MUST be set to zero. The ping functionality does not use the Internal Hop Count field of the reply. (See Section 5.2.1.2)

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

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

The transmitting RBridge MUST wait for a reply frame until a time-out occurs. At that time, the RBridge MUST assume the frame was lost, and this MUST be indicated to the operator. The length of this time-out is not specified in this document.

4.1.2.1. Ping Example

Figure 4 contains a campus with three RBridges. Consider a ping from RB0 to RB2.

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

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

Ping Example Topology

Figure 4

In this diagram RB0 transmits frame (1) destined to RB2. This frame contains the echo request message. When RB1 receives this frame it forwards it to RB2. When RB2 receives this frame it transmits and echo reply frame (2) destined to RB0. RB1 receives this frame and forwards it to RB0.

Below are some select fields for the frames:

Frame #	Ingress RBridge	Egress RBridge	TRILL OAM Protocol	Sequence Number
(1)	RB0	RB2	Echo Request	1
(2)	RB2	RB0	Echo Reply	1

Table 4: Ping Example Frames

For example, if the nicknames for RB0, RB1, and RB2 are 0x0001, 0x0002, and 0x0003 respectively, the console output from such a ping might be:

```
Pinging
--------------------------------------------
... from 0x0001 to 0x0003... 0x0003 is alive
... from 0x0001 to 0x0003... 0x0003 is alive
... from 0x0001 to 0x0003... 0x0003 is alive
```

Table 5: Ping Example Output

In this example, the ping was repeated three times with the sequence number being changed each time.

4.2. Error Reporting

Errors can occur through the reception of TRILL data frames. For this purpose, the error notification format is specified. These are generated due to various events as specified subsequently. When a TRILL data frame is received with an error, an error notification frame MAY be generated. See Section 8 for reasons on some RBridges are allowed to choose not to respond to a request. The generated reply MUST contain the error notification. The sub-code MUST contain a code specifying the error encountered. The valid values are specified in Section 5.2.2.1. Two of these sub-codes contain TLVs with additional information. The error notification also contains a 3 bit error type field which describes the error.

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

When the original ingress RBridge receives the error frame, at a minimum, the RBridge SHOULD update a counter specifying the number of error frames received for the causing error. The encapsulated frame MUST NOT be decapsulated and transmitted. The RBridge SHOULD also keep a set of counters for errors reported by other RBridges.

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

4.2.1. Hop Count Zero Error

When a TRILL data frame is received with a hop-count of zero, an error notification frame MAY be generated. The generated reply MUST contain the hop-count zero error sub-code. If the received frame has the echo request message, the hop-count zero error notification MUST have a sequence number matching the echo request. Otherwise, the sequence number MUST be set to zero. The incoming port ID MUST be the port ID the received frame arrived on. The outgoing port ID MUST be the port ID of the port the received frame would have been forwarded onto if the hop-count was not zero. Finally, the error notification MUST include the 16-bit nickname of the next hop RBridge the frame would have been sent to. If the request is a multi-destination frame, this field MUST be set to all zeros on transmission and ignored on reception. If the RBridge transmitting the request is the egress RBridge, this field MUST be set to 0x0000.

4.2.2. MTU Error

When a TRILL data frame is received with a payload that would exceed the MTU of the port the frame would otherwise be forwarded to, an error notification frame MAY be generated. The generated reply MUST contain the MTU error sub-code. The outgoing port MTU field MUST have the MTU of the port the frame would have otherwise been transmitted on. The incoming port ID MUST be the port ID the received frame arrived on. The outgoing port ID MUST be the port ID of the port the received frame would have been forwarded onto if the frame size was not too large. Finally, the error notification message MUST include the 16-bit nickname of the next hop RBridge the frame would have been sent to. If this is a multi-destination frame this field MUST be set to all zeros on transmission and ignored on reception. If the RBridge transmitting the request is the egress RBridge, this field MUST be set to 0x0000.

5. TRILL OAM Message Format

This section specifies the format of the TRILL OAM message on the wire beyond the ethertype as encoded in the OAM Channel

```
| 0  1  2  3  4  5  6  7| 8| 9|     10-15       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  Version  |         TRILL OAM Protocol        |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|SL|MH|            Flags            |    ERR    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                   Sequence                    |
|                    Number                     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

TRILL OAM Message Common Initial Part

Figure 5

The message fields and flags are as follows:

- o Version, TRILL OAM Protocol, SL, MH, Flags, and ERR: The usage is specified in [I-D.eastlake-trill-rbridge-channel]. The SL bit SHOULD be 0. The MH bit MUST be 1. The version must be 0. ERR MUST be all zeros. The TRILL OAM Protocol is further specified by the tool type.

- o Sequence Number: This field is used to sequence frames for certain tools. Not all tools utilize the sequence number field.

5.1. Protocol Code Values

The protocol code values which specify the tool type are:

- o 0x004 (Suggested): Echo Request, See Section 5.2.1.1

- o 0x005 (Suggested): Echo Reply, See Section 5.2.1.2

- o 0x006 (Suggested): Error Notification, See Section 5.2.2

5.2. Protocol Codes Formats

5.2.1. Protocol Application Codes Formats

5.2.1.1. Echo Request

```
| 0  1  2  3  4  5  6  7| 8| 9|     10-15       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  Version  |         TRILL OAM Protocol        |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|SL|MH|           Flags            |    ERR    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                   Sequence                    |
|                    Number                     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Echo Request

Figure 6

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

o Version, TRILL OAM Protocol, SL, MH, Flags, and ERR: The usage is specified in [I-D.eastlake-trill-rbridge-channel]. The SL bit SHOULD be 0. The MH bit MUST be 1. The version must be 0. ERR MUST be all zeros. TRILL OAM Protocol MUST be 0x004 (Suggested).

o Sequence Number: An arbitrary 32-bit unsigned integer used to aid in matching reply messages to echo requests. MAY be zero.

5.2.1.2. Echo Reply

```
| 0  1  2  3  4  5  6  7| 8| 9|     10-15       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  Version  |         TRILL OAM Protocol        |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|SL|MH|           Flags              |    ERR    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                   Sequence                    |
|                    Number                     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|             Reserved          | I. Hop Count |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
.                                               .
.                     TLVs                      .
.                                               .
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Echo Reply Format

Figure 7

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

o Version, TRILL OAM Protocol, SL, MH, Flags, and ERR: The usage is specified in [I-D.eastlake-trill-rbridge-channel]. The SL bit SHOULD be 0. The MH bit MUST be 1. The version must be 0. ERR MUST be all zeros. TRILL OAM Protocol MUST be 0x005 (Suggested).

o Reserved: A reserved field. Set to zero on transmission and ignored on reception.

o Internal Hop Count: If the request being replied to was an echo request, this value MUST be zero on transmission and ignored on reception. If the request being replied to was a respond request, this value is a copy of the TRILL Hop Count value in the request.

The reserved and internal hop-count fields combined occupy the subcode field of the TRILL OAM message.

o Sequence Number: A 32-bit unsigned integer used to aid in matching reply messages to echo requests. This MUST match the request being replied to.

o TLVs: A set of type, length, value encoded fields as specified in Section 5.3. The next hop nickname, outgoing port ID, and incoming port ID TLVs are required.

5.2.2. Error Notification Format

```
| 0  1  2  3  4  5  6  7| 8| 9|     10-15       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  Version  |         TRILL OAM Protocol        |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|SL|MH|           Flags             |    ERR    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                   Sequence                    |
|                    Number                     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Err. T.|                Subcode               |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
.                                               .
.                     TLVs                      .
.                                               .
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Error Format

Figure 8

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

o Version, TRILL OAM Protocol, SL, MH, Flags, and ERR: The usage is specified in [I-D.eastlake-trill-rbridge-channel]. The SL bit SHOULD be 0. The MH bit MUST be 1. The version must be 0. ERR MUST be all zeros. TRILL OAM Protocol MUST be 0x006 (Suggested).

o Sequence Number: For all sub-codes except for the hop count error this field is unused. It is set to zero on transmission and ignored on reception. For the hop count error this is a 32-bit unsigned integer used to aid in matching reply messages to echo requests requests. If the frame whose hop-count dropped to zero contains the echo request message (See Section 5.2.1.1), this MUST match the sequence number echo request found in that message. If this is not in reply to a request, then the sequence number MUST be set to zero.

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

- o Subcode: MUST be a specifier of the error discovered in the frame. The valid values are specified in Section 5.2.2.1

- o TLVs: A set of type, length, value encoded fields as specified in Section 5.3. For next hop errors the next hop nickname, outgoing port ID, and incoming port ID TLVs MUST be present. For MTU errors the outgoing port MTU, next hop nickname, outgoing port ID, and incoming port ID TLVs MUST be present. For all other errors the TLVs are not used and the length of this section is set to zero.

5.2.2.1. Error Specifiers

The sub-code values fall into three categories: errors, warnings, and comments. All sub-codes represent something out of the ordinary that has gone wrong, but certain ones are more important than others. Sub-codes that are classified as errors are the most severe with warning sub-codes being slightly less severe. These are enabled by default. Sub-codes classified as comments are minor and are disabled by default. They may be useful for operators debugging a network. All error generations are optional and therefore MAY be generated or not generated depending on security and implementation constraints.

The error specifiers sub-code values are:

Sub-codes

- o 0: Unknown Error: Indicates an error has occurred.

- o 1: Corrupt Frame: Frame received with invalid FCS or that was not an 8-bit multiple in length. It could be impossible for a device to signal this if the low-level port hardware hides this from the software.

- o 2: Invalid Outer.MacDA: Indicates the MAC Address is a multicast address and the M bit is zero, the MAC Address is not a multicast address and the M bit is one, or the M bit is zero and the frame carried is an ESADI frame.

- 3: Illegal Outer.VLAN: Indicates the Outer.VLAN ID is 0xFFF.
- 4: Invalid Outer.VLAN: Indicates the Outer.VLAN ID was not the designated VLAN ID.
- 5: Unknown TRILL Version: Indicates the TRILL Version is unknown.
- 6: Op-Length Exceeds Frame Length: Indicates the Op-Length says the options field extends beyond the end of the received frame length.
- 8: Unknown Egress RBridge: Indicates the Egress RBridge in a received frame is unknown.
- 9: Unknown Ingress RBridge: Indicates the Ingress RBridge in a received frame is unknown.
- 10: Unsupported Critical Hop-by-hop Option: Indicates an unsupported critical hop-by-hop option was received.
- 11: Unsupported Critical Ingress-to-Egress Option: Indicates an unsupported critical ingress-to-egress option was received.
- 12-84: Available for allocation by IETF Review
- 85: Reserved for Private Experimentation

Warning Sub-codes

- 86: Illegal Inner.VLAN: Indicates the Inner.VLAN ID is 0xFFF.
- 87: Inner/Outer VLAN Priority Mismatch: Indicates the priority values in the inner and outer VLANs do not match.
- 88: P2P Hello on TRILL Hello Link: Indicates a P2P Hello was received on a TRILL Hello Link.
- 89: TRILL Hello on P2P Hello Link: Indicates a TRILL Hello was received on a P2P Hello Link.
- 90: No Adjacency: Indicates a TRILL data frame was sent from an RBridge the receiving RBridge is not adjacent to.
- 91: Encapsulated BPDU/VRP Frame: A TRILL Frame containing a BPDU or VRP frame was received.
- 92: Invalid Mutability Flag: Indicates the mutability flag was set on a received CHbH Option.

- 93: Invalid TLV Option Length: Indicates the option length field of a TLV option was between 121 and 127.

- 94: Options Ordering Error: Indicates the TLV options are ordered incorrectly.

- 95: Additional Flag TLV Zero: Indicates a problem in the additional Flag TLV.

- 96: Configured Nickname Collision: Indicates an RBridge was detected in the campus with the same nickname (Configured or not).

- 97: Multiple DRBs detected.

- 98: Multiple appointed forwarders detected.

- 99-169: Available for allocation by IETF Review

- 170: Reserved for Private Experimentation

Comment Sub-codes

- 171: Inner.VLAN C-Bit Set: Indicates the C-Bit in the Inner.VLAN is set.

- 172: Unknown Inner.MacDA: Indicates the Inner.MacDA is unknown. This may occur if devices are configured to explicitly register end stations and an unknown Inner.MacDA occurs in a unicast TRILL data frame. This also only applies at egress and could indicate that the Inner.MacDA was a learned address that has timed out.

- 173: Unknown Inner.MacSA: Indicates the Inner.MacSA is unknown. This may occur if devices are configured to explicitly register end stations and an unknown Inner.MacSA occurs in a TRILL data frame.

- 174: Outer.VLAN C-Bit Set: Indicates the C-Bit in the Outer.VLAN is set for an Ethernet frame.

- 175: Invalid Reserved Bits: Indicates the reserved bits are non-zero in a received frame.

- 176: Invalid Nickname: Indicates a nickname in the reserved space of 0xFFC0 to 0xFFFF was received that is not implemented at the receiving RBridge.

- 177: Unsupported Non-Critical Hop-by-hop Option: Indicates an unsupported non-critical hop-by-hop option was received. While sending a non-critical option to an unsupported device is not an error, this could be used to support identification of devices needing an upgrade.

- 178: Unsupported Non-Critical Ingress-to-Egress Option: Indicates an unsupported non-critical ingress-to-egress option was received. While sending a non-critical option to an unsupported device is not an error, this could be used to support identification of devices needing an upgrade.

- 179: Performance Exceeded: Indicates a frame was discarded due to performance problems such as a buffer overflow.

- 180: Insufficient Hop Count: Indicates a frame was received with a hop-count that was insufficient to reach the destination.

- 181-254: Available for allocation by IETF Review

- 255: Reserved for Private Experimentation

5.3. Type, Length, Value (TLV) Encodings

To facilitate future interoperable expansion of the data carried in OAM sub-messages some sub-messages use a TLV encoding. These TLV sections consist of a list of type, length, value encoded data where the type signals to the RBridge how to interpret the value, and the length tells the RBridge the length of the value in bytes. The type and length are both 8 bit fields. A length of zero indicates the value is a UTF-8 string with a NULL ('\0') terminating byte. Preceeding the list of TLVs is a 16 bit total length field which specifies the total length of all the length fields in octet units.

```
| 0  1  2  3  4  5  6  7| 8| 9|     10-15       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                  Total Length                 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
.                                               .
.                   TLV List                    .
.                                               .
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

TLVs Format

Figure 9

Each TLV in the TLV List appears on the wire as such:

```
| 0  1  2  3  4  5  6  7| 8| 9|     10-15       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|         Type          |         Length        |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
.                                               .
.                     Value                     .
.                                               .
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

TLV Format

Figure 10

The type values are:

- o 0: Next Hop Nickname, See Section 5.3.1.1
- o 1: Outgoing Port ID, See Section 5.3.1.3
- o 2: Incoming Port ID, See Section 5.3.1.2
- o 3: Outgoing Port MTU, See Section 5.3.1.4
- o 4-253: Available for allocation by IETF Review
- o 254: Reserved for Private Use
- o 255: Reserved

5.3.1. TLV Types

5.3.1.1. Next Hop Nickname

```
| 0  1  2  3  4  5  6  7| 8| 9|     10-15       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|     Type = 0x01       |     Length = 0x02     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|               Next Hop Nickname               |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Next Hop Nickname Format

Figure 11

For traceroutes targeting known unicast destinations, hop-count errors, and MTU errors, this TLV MUST be the 16-bit nickname of the next hop RBridge the frame is being or would have been sent to. If the RBridge transmitting the TLV is the egress RBridge this field MUST be set to 0x0000. For traceroutes targeting multi-destination destinations, e.g. with the TRILL M bit high, this field contains the nickname of the RBridge the frame being responded to is from. For pings, this field MUST be set to all zeros on transmission and ignored on reception. For multi-destination hop-count errors this field contains the nickname of the RBridge the frame with the exceeded hop-count was sent from. For multi-destination MTU error traffic, this field MUST be set to all zeros on transmission and ignored on reception.

5.3.1.2. Incoming Port ID

```
 | 0  1  2  3  4  5  6  7| 8| 9|     10-15       |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |     Type = 0x02       |      Length = 0x02    |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |                Incoming Port ID               |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Incoming Port ID Format

Figure 12

This TLV MUST be set to the Port ID found in 'The Special VLANs and Flags sub-TLV' for the port the request being replied to was received on. ([I-D.ietf-isis-trill])

5.3.1.3. Outgoing Port ID

```
 | 0  1  2  3  4  5  6  7| 8| 9|     10-15       |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |     Type = 0x03       |      Length = 0x02    |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |                Outgoing Port ID               |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Outgoing Port ID Format

Figure 13

This TLV MUST be set to the Port ID found in 'The Special VLANs and

Flags sub-TLV' for the port the frame is being forwarded on to (or would have been for an echo request/hop-count error). ([I-D.ietf-isis-trill]) If the request was consumed by the replying RBridge, the port ID MUST be 0xFFFF.

5.3.1.4. Outgoing Port MTU

```
| 0  1  2  3  4  5  6  7| 8| 9|     10-15       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|     Type = 0x04       |      Length = 0x02    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|               Outgoing Port MTU               |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Outgoing Port MTU Format

Figure 14

This TLV MUST be the MTU of the outgoing port specified in the outgoing port ID TLV.

6. Acknowledgments

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

7. IANA Considerations

IANA is request to create a new subregistry within the TRILL Parameter registry for "TRILL OAM Message Error Sub-Message Error Specifiers". This subregistry that is initially populated as specified in Section 5.2.2.1. Additional values are allocated by IETF Review [RFC5226].

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

This draft also requires action to reserve the TRILL OAM Control Channel protocol codes.IANA is requested to allocate the TRILL OAM Channel protocol codes for as listed in Section 5.1.

8. Security Considerations

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

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

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

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

9. References

9.1. Normative References

[I-D.eastlake-trill-rbridge-channel] 3rd, D., Manral, V., Ward, D., Yizhou, L., and S. Aldrin, "RBridges: OAM Channel Support in TRILL", draft-eastlake-trill-rbridge-channel-00 (work in progress), March 2011.

[I-D.ietf-isis-trill] 3rd, D., Banerjee, A., Dutt, D., Perlman, R., and A. Ghanwani, "TRILL Use of IS-IS", draft-ietf-isis-trill-05 (work in progress), February 2011.

[I-D.ietf-trill-rbridge-protocol] 3rd, D., Dutt, D., Gai, S., Ghanwani, A., and R. Perlman, "Rbridges: Base Protocol Specification", draft-ietf-trill-rbridge-protocol-16 (work in progress), March 2010.

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

9.2. Informative References

[I-D.ietf-trill-rbridge-mib] Rijhsinghani, A. and K. Zebrose, "Definitions of Managed Objects for RBridges", draft-ietf-trill-rbridge-mib-02 (work in progress), March 2011.

[IEEE.802-1ag] Institute of Electrical and Electronics Engineers, "IEEE Stadard for Local and metropolitian area networks / Virtual Bridged Local Area Networks / Connectivity Fault Management", IEEE Standard 802.1Q, December 2007.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in

RFCs", BCP 26, RFC 5226, May 2008.

[RFC5837] Atlas, A., Bonica, R., Pignataro, C., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, April 2010.

Appendix A. Revision History

RFC Editor: Please delete this appendix before publication.

A.1. Changes from -00 to -01

Reworked the document to use the OAM Channel rather than an OAM option.

Changed the frame formats to work within the OAM Channel.

Numerous minor typo corrections and wording clarifications.

Removed the route-respond traceroute.

Combined all the error notifications into one OAM Channel.

Authors' Addresses

David Michael Bond
University of New Hampshire InterOperability Laboratory
121 Technology Drive Suite #2
Durham, New Hampshire 03824
US

Phone: +1-603-339-7575
EMail: david.bond@iol.unh.edu
URI: http://mokon.net

Vishwas Manral
IP Infusion Inc.
1188 E. Arques Ave.
Sunnyvale, CA 94089
US

EMail: vishwas@ipinfusion.com

TRILL working group L. Dunbar
Internet Draft Huawei
Intended status: Standard Track
Expires: Sept 2011
March 7, 2011

Directory Server Assisted TRILL edge
draft-dunbar-trill-server-assisted-edge-00.txt

Status of this Memo

Copyright Notice

Abstract

TRILL edge nodes currently learn the mapping between MAC address and its corresponding TRILL edge node address by observing the data packets traversed through.

This document describes why and how directory based server(s) can optimize TRILL network in data center environment.

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.

Table of Contents

1. Introduction

Data center networks are different from campus networks in several ways. Main differences include:

VM (host) to server assignment is done by Server (or VM) Manager, which means that the host location is arranged by management system(s).

Topology is based on racks and rows;

There could be massive number of virtual machines (hosts), but relatively small number of switches.

This draft describes why Data Center TRILL networks can be optimized by utilizing directory server based approach.

2. Terminology

Bridge: IEEE802.1Q compliant device. In this draft, Bridge is used interchangeably with Layer 2 switch.

DC: Data Center

EOR: End of Row switches in data center.

FDB: Filtering Database for Bridge or Layer 2 switch

ToR: Top of Rack Switch. It is also known as access switch.

VM: Virtual Machines

3. Impact to TRILL by massive number of hosts

In a virtualized data center, a VM may be placed on any physical server. A variety of algorithms can be applied to select the location of a VM. Resource aware algorithms (e.g. energy, bandwidth, etc,) will use a placement that satisfies the processing requirements of each VM but require the minimal number of physical servers and switching devices.

With this, and similar types of assignment algorithm, subnets tend to extend throughout the network. When this happens, the broadcast messages within each subnet will be flooded across the TRILL domain, which not only consumes a lot of bandwidth on links in TRILL domain, but also causes a TRILL edge port to learn all the hosts belonging to all the subnets which are enabled on the port. Even though a TRILL edge port is only supposed to learn the entries which communicate with hosts underneath, the frequent ARP/ND from all hosts within each subnet will always refresh the TRILL edge node’s MAC<->TRILL-Edge mapping table.

Consider a data center with 80 rows, 8 racks per row and 40 servers per rack. There can be 80*8*40=25600 servers. Suppose each server is virtualized to 20 VMs, there could be 25600*20=512000 hosts in this data center.

Let’s consider a case that the TRILL edge starts at an Ingress port of a TOR switch. Assuming there are 5 different VLANs enabled on the TRILL Ingress port (i.e. the 20 VMs in one server belong to 5 different VLANs) and each VLAN has 200 hosts, then the TRILL edge port has to learn 5*200=1000 MAC&VLAN entries. Since there are 40 ports on the TOR, the total number of MAC&VLAN entries for the TOR switch is 1000*40= 40000. Under this scenario, there will be 25600 entries in the TRILL routing domain if protection is not considered. When protection is considered, the number of ports in TRILL domain will double. That may be too many nodes for the IS/IS routing domain. Let’s consider another case of TRILL edge starting at the End of Row switches. With the same assumption as before, there are 40*20 = 800 hosts to attached to each port of an EoR switch and 8*800=6400 hosts attached to an EoR switch. If all those 6400 hosts belong to 640 VLANs and each VLAN has 200 hosts, then the total number of MAC&VLAN entries to be learned by the TRILL edge (i.e. EoR) = 640*200=128000. Under this scenario, there will be 80*8 = 640 EoR ports in the TRILL routing domain when protection is not considered and 1280 EoR ports when protection is considered. However, the number of MAC&VLAN entries to be learnt by the TRILL edge node is very large.

4. Directory Server for TRILL in Data Center environment.

As described in the Section 1, the VM placement to server/rack is orchestrated by Server (or VM) Management System(s). Therefore, there is a central location with the information on where each VM is placed. So it is relatively reliable to build a centralized (or distributed) directory server(s) who has the knowledge on where each VM is placed.

Here can be a procedure for TRILL edge node to utilize a Directory Server

TRILL edge node can simply intercept all ARP requests and forward them to the Directory Server,

The reply from the Directory Server can be the standard ARP reply with an extra field showing the TRILL egress node address

TRILL ingress node can cache the mapping

If TRILL edge node receives an unknown MAC-DA, it simply forwards the packet to the directory server. The directory server can simply drop the frame if it doesn’t have the information, or forward the frame to the correct egress node and send down a new mapping to the ingress Trill edge node.

Another approach is for Directory Server to pass down the MAC&VLAN mapping for all the hosts belonging to all the VLANs enabled on the TRILL edge port.

5. Conclusion and Recommendation

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

Therefore, we suggest TRILL to consider directory assisted approach(es). This draft only introduces the basic concept of using directory assisted approach for TRILL edge nodes to learn the MAC to TRILL mapping. We want to get some working group consensus before drilling down to detailed steps required for the approach.

6. Manageability Considerations

This document does not add additional manageability considerations.

7. Security Considerations

This document has no additional requirement for security.

8. IANA Considerations

9. Acknowledgments

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

10. References

[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

Intellectual Property Statement

Disclaimer of Validity

Acknowledgment

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

TRILL Working Group
INTERNET-DRAFT
Intended status: Proposed Standard
Updates: RFCtrill

Donald Eastlake
Huawei
Vishwas Manral
IP Infusion
Dave Ward
Juniper
Li Yizhou
Sam Aldrin
Huawei

Expires: September 6, 2011
March 7, 2011

RBridges: OAM Channel Support in TRILL
<draft-eastlake-trill-rbridge-channel-00.txt>

Abstract

This document specifies a general channel for sending OAM (Operations, Administration, and Maintenance) messages in an RBridge campus through an extension to the TRILL (TRansparent Interconnection of Lots of Links) protocol.

Status of This Memo

Table of Contents

1. Introduction

RBridge campuses provide Layer 2 data networking using the TRILL protocol. However, the TRILL base protocol specification [RFCtrill] does not specifically provide for OAM (Operations, Administration, and Maintenance) messages. This document specifies a facility for the transmission of OAM messages within an RBridge campus.

Familiarity with [RFCtrill] is assumed in this document.

1.1 TRILL Channel Requirements

It is anticipated that various OAM protocols operating at the TRILL level will be desired in RBridge campuses. For example, there is a need for rapid response continuity checking with a protocol such as BFD [RFC5880] [RFC5882] and for a variety of optional reporting, in the spirit of some ICMP [RFC792] messages, such as reporting Hop Count exhaustion, unknown egress nickname in the TRILL header, and the like, including ping and trace route functions.

To avoid having to design and specify a way to carry each new OAM protocol in TRILL, this document specifies a general channel for sending OAM messages between RBridges in a campus at the TRILL level using extensions to the TRILL protocol. To accommodate a wide variety of OAM protocols, the OAM Channel facility accommodates all the regular modes of TRILL Data transmission including single and multiple hop unicast as well as VLAN scoped multi-destination distribution. To minimize any unnecessary burden on transit RBridges and to provide a more realistic test of network continuity and the like, TRILL OAM Channel messages are designed to look like TRILL Data frames and, in the case of multi-hop messages, can normally be handled by transit RBridges as if they were TRILL data frames; however, to enable processing of an OAM message at transit RBridges when required, an optional Alert non-critical hop-by-hop extended header flag is specified to cause transit RBridge to examine a frame with that flag set.

This document also provides a format for sending OAM messages between end stations and RBridges, in either direction, when appropriate for the OAM protocol involved.

Each particular OAM protocol will likely use only a subset of the facilities specified herein.

The TRILL OAM Channel is similar to the MPLS Generic Channel specified in [RFC5586]. Instead of using a special MPLS label to indicate a special channel message, a TRILL OAM Channel message is indicated by a special Inner.MacDA.

1.2 Terminology

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

The terminology and acronyms of [RFCtrill] are used in this document with the additions listed below.

BFD - Bidirectional Forwarding Detection

ICMP - Internet Control Message Protocol

MH - Multi-Hop

OAM - Operations, Administration, and Maintenance

OV - OAM (Message Channel) Version

SL - Silent

2. The TRILL OAM Channel Messages

TRILL OAM messages are transmitted as TRILL Data frames. They are identified as OAM messages by their Inner.MacDA. The encapsulated frame has, after the Inner Ethernet Header, the TRILL-OAM Ethertype that is part of an OAM Channel Header. That Header indicates the OAM protocol of the following OAM protocol specific data.

The diagram below shows the overall structure of a TRILL OAM Message Channel frame on a link between two RBridges:

```
            Frame Structure               Section of This Document
                                          ------------------------
   +--------------------------------+
   |       Outer Link Header        |   Section 2.4 if Ethernet Link
   +--------------------------------+
   |          TRILL Header          |   Section 2.2
   +--------------------------------+
   |     Inner Ethernet Header      |   Section 2.1.2
   +--------------------------------+
   |    TRILL OAM Channel Header    |   Section 2.1.1
   +--------------------------------+
   | OAM Protocol Specific Payload  |   See specific OAM protocol
   +--------------------------------+
   | Link Trailer (FCS if Ethernet) |
   +--------------------------------+
```

Some OAM messages may require examination of the frame, to determine if the transit RBridge needs to take any action, by transit RBridges that support the OAM Channel feature. To indicate this, a non-critical hop-by-hop extended TRILL header flag is allocated as the Alert bit, as further described in Section 4 below.

In addition, a TRILL Header extended flag is provided that may optionally be used to guarantee that frames sent over the TRILL OAM Message Channel cannot be accidentally forwarded to end stations, even by minimally conformant RBridges that are ignorant of the TRILL OAM Message Channel feature.

The Sections 2.1 and 2.2 below describe the Inner frame and the TRILL Header for frames sent in the TRILL OAM Message Channel. As always, the Outer link header is whatever is needed to get a TRILL Data frame to the next hop RBridge, depends on link technology, and can change with each hop for multi-hop OAM messages. Section 2.4 describes the Outer link header for Ethernet. And Section 2.5 discusses some special considerations for the first hop transmission of OAM Channel messages.

Section 3 describes the OAM-Channel extended flag. Section 4 describes some details of TRILL OAM Message processing. And Section 5 specifies an optional format for native OAM frames.

2.1 The OAM Message Inner Frame

The encapsulated Inner frame within a TRILL OAM Message Channel frame is as shown below.

```
 Inner Ethernet Header:
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Special Inner.MacDA                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Special Inner.MacDA cont.   |         Inner.MacSA             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Inner.MacSA cont.                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Ethertype = C-Tag (0x8100)    |   Priority, VLAN ID             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 TRILL OAM Channel Header:
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |       TRILL-OAM Ethertype     |  OV   |  TRILL OAM Protocol     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Flags       |  ERR  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 OAM Protocol Specific Information:
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                                 |
   +                   OAM Protocol Specific Data
   |  ...
```

The OAM protocol specific data contains the information related to the specific protocol type used in the OAM channel message. Details of that data are outside the scope of the document, except in the case of the OAM Channel error protocol specified below.

2.1.1 TRILL OAM Channel Header

As shown in the diagram above, the TRILL OAM header starts with the TRILL OAM Ethertype (see Section 6.2). Following that is a four-byte quantity with four sub-fields as follows:

OV gives the OAM Header version and MUST be zero.

A 12-bit field that specifies the particular TRILL OAM protocol to which the message applies.

Flags provides 12 bits of flags described below.

ERR is a four-bit field used in connection with error reporting at the OAM Channel level as described in Section 4.

The flag bits are numbered from 0 to 11 as shown below.

```
  0  1  2  3  4  5  6  7  8  9 10 11
+--+--+--+--+--+--+--+--+--+--+--+--+
|SL|MH|       Available Flags       |
+--+--+--+--+--+--+--+--+--+--+--+--+
```

Bit 0, which is the high order bit in network order, is defined as the SL or Silent bit. If it is a one, it suppresses OAM Channel Error messages (see Section 4).

Bit 1 is the MH or Multi-Hop bit. It is used to inform the destination OAM protocol that the message was intended to be multi-hop (MH=1) or one-hop (MH=0).

The TRILL OAM Protocol field specifies the OAM protocol that the OAM Channel message relates to. The initial defined value is listed below. See Section 5 for IANA Considerations.

Protocol	Name - Section of this Document
0x0001	OAM Channel Error - Section 4

2.1.2 Inner Ethernet Header

The special Inner.MacDA is All-OAM-RBridges to signal that the frame is a TRILL OAM Chanel message (see Section 6.1).

The RBridge originating the OAM message selects the Inner.MacSA. Because OAM Channel messages are handled very much like ordinary TRILL Data frames, if the Inner.MacSA is a unicast MAC address, on decapsulation it will be learned as being attached to the ingress RBridge. If that learning is not desired, the Inner.MacSA MAY be set to All-OAM-RBridges or the like. MAC address learning on does not occur if the MAC address has the group bit on.

2.1.3 Inner.VLAN

As with all TRILL encapsulated frames, a VLAN tag MUST be present. Use of a VLAN tag Ethertype other than 0x8100 or stacked VLAN tags is beyond the scope of this document.

Multi-destination TRILL OAM messages are, like all multi-destination TRILL Data messages, VLAN scoped so the Inner.VLAN ID MUST be set to the VLAN of interest. To the extent that distribution tree pruning is in effect, such OAM messages will only reach RBridges advertising that they have appointed forwarder connectivity to that VLAN.

For known unicast OAM messages, if the message is one-hop it is RECOMMENDED that the Inner.VLAN ID be the Designated VLAN on that hop. For multi-hop unicast OAM messages, it is RECOMMENDED that the Inner.VLAN ID be the default VLAN 1.

The Inner.VLAN will specify a three-bit frame priority for which the following recommendations apply:

- For one-hop OAM messages critical to network connectivity, such as one-hop BFD for rapid link failure detection in support of TRILL IS-IS, the RECOMMENDED priority is 7.

- For single and multi-hop known unicast OAM messages important to network operation but not critical for connectivity, the RECOMMENDED priority is 6.

- For other known unicast OAM messages and all multi-destination OAM messages, it is RECOMMENDED that the default priority zero be used and, in any case priorities higher than 5 SHOULD NOT be used.

2.3 The TRILL Header for OAM Messages

After the Outer link header (which, for Ethernet, ends with the TRILL Ethertype) and before the encapsulated frame, the OAM message’s TRILL Header appears as follows:

```
                             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                             |V=0| R |M| Op-Len  | Hops=0x3F |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |       Egress Nickname          |       Ingress Nickname       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The TRILL Header version V MUST be zero, the R bit are reserved, the M bit is set appropriately as the OAM message is known unicast (M=0) or multi-destination (M=1), and Op-Len is set appropriately for the length of the options area, if any, all as specified in [RFCtrill].

When a TRILL OAM message is originated, the hop count field MUST be set to the maximum value, 0x3F. For messages sent a known number of hops, particularly one-hop messages or two-hop neighbor echo messages, checking the Hops (Hop Count) field provides an additional validity check as discussed in [RFC5082].

The RBridge originating a TRILL OAM message places a nickname that it holds into the ingress nickname field.

There are several cases for the egress nickname field. If the OAM message is multi-destination, then the egress nickname designates the distribution tree to use. If the OAM message is a multi-hop unicast message, then the egress nickname is a nickname of the target RBridge; this includes the special case of an "echo" OAM message where the originator places one of its own nicknames in both the ingress and egress nickname fields. If the OAM message is a one-hop unicast message, there are two possibilities for the egress nickname.

- o The egress nickname can bet set to a nickname of the target neighbor RBridge.

- o The special nickname Any-RBridge may be used. RBridges supporting the TRILL OAM Channel facility MUST recognize the Any-RBridge special nickname and accept TRILL Data frames having that value in the egress nickname field as being sent to them as the egress. Thus, for such RBridges, using this egress nickname guarantees processing by an immediate neighbor regardless of the state of nicknames.

2.4 OAM Message Ethernet Link Header

If the link on which a TRILL OAM frame is transmitted between neighbor RBridges is Ethernet, the link header follows the usual rules for a TRILL Data frame over Ethernet [RFCtrill]. In particular, the Outer.MacSA is the MAC address of the port from which the frame is sent. The Outer.MacDA is the MAC address of the next-hop RBridge port for unicast TRILL OAM messages or the All-RBridges multicast address for multi-destination TRILL OAM messages. The Outer.VLAN tag specifies the Designated VLAN for that hop and the priority must be the same as in the Inner.VLAN tag; however, the output port may have been configured to strip VLAN tags, in which case no Outer.VLAN tag appears on the wire.

2.5 Special Transmission and Rate Considerations

If a multi-hop OAM Channel message is received by an RBridge, the criteria and method of forwarding it is the same as for any TRILL Data frame. If it is so forwarded, it will be on a link that was included in the routing topology because it was in Report state as specified in [RFCadj].

However, special considerations apply to the first hop because it may be desirable to use some OAM messages on links that are not yet fully up. In particular, it is permissible, if specified by the particular OAM protocol, for the source RBridge that has created an OAM Channel message to transmit it to a next hop RBridge when the link is in the Detect and Two-Way states, as specified in [RFCadj], as well as when it is in the Report state.

OAM messages may represent a burden on the RBridges in a campus and should be rate limited, especially if they are multi-destination, multi-hop, and/or have the Alert extended flag set.

3. The TRILL OAM-Channel Extended Flag

If an OAM Channel ignorant RBridge were to receive an OAM Channel frame, it would generally flood the encapsulated frame out all ports where it was the appointed forwarder for the frame’s VLAN as specified by the Inner.VLAN ID. It may be desirable to stop such flooding in case, due to transient conditions, an OAM Channel frame is misdelivered to an OAM Channel ignorant RBridge. It is also desirable for an RBridge to be able to indicate that it supports the OAM Channel facility.

To provide these facilities, a critical ingress-to-egress TRILL Header extended flag, OAM-Channel, is specified for the TRILL OAM Channel facility [TRILLopt]. This flag is not required to be set in the TRILL Header in TRILL OAM message frames. It serves the two functions described above, as follows:

- An RBridge indicates that it supports the TRILL OAM Channel facility by advertising, in the link state database, its support for this extended flag.

- If this extended flag is set in a TRILL OAM message frame, it guarantees that, if the inner frame is processed for egress by an RBridge that does not implement the TRILL OAM Channel, the decapsulated frame will be discarded because egress RBridges are required by the base standard to discard frames indicating a critical ingress-to-egress extended flag they do not support. If it is certain that all RBridges in the campus implement the TRILL OAM Channel or if the possible local flooding of the inner frame as described above is acceptable, there is no requirement to include an options area nor to set this particular extended flag in the TRILL Header even if an options area is included.

As with any other critical ingress-to-egress extended flag, if this extended flag is set, then the summary CItE bit MUST be set at the top of the options area.

4. Processing TRILL OAM Chanel Messages

TRILL OAM messages are designed to look like and, to the extent practical, be processed as regular TRILL Data frames. On receiving a TRILL OAM frame, the initial tests on the Outer.MacDA, Outer Ethertype, TRILL Header V and Hop Count fields and the Reverse Path Forwarding Check if the frame is multi-destination, are all performed as usual. The forwarding and/or decapsulation decisions are the same as for a regular TRILL Data frame with following exceptions for RBridges implementing the TRILL OAM Channel:

1. An RBridge implementing the TRILL OAM Channel MUST recognize the Any-RBridge egress nickname in unicast TRILL Data frames, decapsulating and not forwarding such frames if they meet other checks.

2. If the Alert extended flag is set, then the RBridge needs to process the OAM Channel message as described below even if it is not egressing the frame. If it is egressing the frame, then no additional processing beyond egress processing is needed even if the Alert flag is set.

3. On decapsulation, the special Inner.MacDA value of All-OAM-RBridges MUST be recognized to trigger processing as a TRILL OAM Channel message.

If the OAM-Channel extended flag is present and set and an egressing RBridge does not implement the TRILL OAM Channel feature, the frame is discarded. If other extended flags or options are present, they may affect processing or cause the frame to be discarded.

4.1 Processing the TRILL OAM Channel Header

Knowing that it has a TRILL OAM Channel message, the egress RBridge, and any transit RBridge if the Alert bit is set in the TRILL Header, looks at the OV (OAM Message Header version) and OAM Protocol fields; however, if the frame is so short that the Ethertype or the OAM Channel Header does not fit or the Ethertype is other than TRILL-OAM, the frame is discarded.

If any of the following conditions occur at an egress RBridge, the frame is not processed and an error may be generated as specified in Section 4.2; however, if these conditions are detected at a transit RBridge examining the message because the Alert flag is set, no error is generated and the frame is still forwarded normally.

1. The OV field is non-zero.

2. The OAM Protocol field is a reserved value or a value unknown to the processing RBridge.

3. The ERR field is non-zero and OAM protocol is a value other than 0x001.

If the OV field is zero and the processing RBridge recognizes the OAM Protocol value, it processes the message in accordance with that OAM protocol. The processing model is as if the received frame starting with and including the TRILL Header is delivered to the OAM protocol along with a flag indicating whether this is (a) transit RBridge processing due to the Alert flag being set or (b) egress processing.

Errors within a recognized OAM Protocol are handled by that OAM protocol itself and do not produce OAM Message Channel Error frames.

4.2 OAM Channel Errors

A variety of problems at the OAM Channel level cause the return of an OAM Channel Error frame unless the "SL" (Silent) flag is a one in the OAM message for which the problem was detected or the frame in error appears, itself, to be an OAM Channel error frame or the error is suppressed due to rate limiting.

An OAM Channel Error frame is a multi-hop unicast TRILL OAM Channel message with the ingress nickname set to the nickname of the RBridge detecting the error, and the egress nickname set to the value of the ingress nickname in the OAM message for which the error was detected. The SL and MH flags SHOULD be set to one and the ERR field MUST be non-zero as described below. In case more than one error applies, the lower numbered ERR value is used. For the protocol specific data area, an OAM Channel Message Error frame has at least the first 256 bytes (or less if less are available) of the erroneous decapsulated OAM message starting with the TRILL Header.

The following values for ERR are specified:

ERR	Meaning
0	- Not an OAM Channel error frame.
1	Unimplemented value of OV
2	Reserved or unimplemented value of Protocol
3	ERR field is non-zero but Protocol field does not equal 0x001
4-15	- Available for allocation, see Section 6.1.

All RBridges implementing the TRILL OAM Message Channel feature MUST recognize the OAM Message Channel Error protocol value (0x001). They MUST NOT generate an OAM Message Channel Error message in response to a TRILL OAM Channel Error message, that is an OAM message with a protocol value of 0x001.

5. Native TRILL-OAM Frames

If provided for by the OAM protocol involved, native TRILL OAM messages may be sent between end-stations and RBridges in either direction. Such native frames have the TRILL-OAM Ethertype and look like the encapsulated frame within a TRILL OAM Channel message with the following exceptions:

1. TRILL does not require the presence of VLAN tagging on such native TRILL OAM frames. However, port configuration, link characteristics, or the OAM protocol involved may require such tagging.

2. If the frame is unicast, the destination MAC address is the unicast MAC address of the RBridge or end-station port that is its intended destination. If the frame is multicast to all the RBridges on a link that support some OAM protocol that uses this transport, the destination MAC address is All-OAM-RBridges. If the frame is multicast to all the devices that TRILL considers to be end stations on a link that support some OAM protocol that uses this transport, the destination MAC address is TRILL-End-Stations (see Section 6.1).

3. As with any native frame, the source MAC address is that of the port sending the frame.

A native frame with the TRILL-OAM Ethertype must meet the usual VLAN and destination MAC address restrictions to be accepted by an RBridge. If provided for by the OAM protocol involved, the receipt of such a native frame MAY lead to the generation and forwarding of one or more TRILL OAM Channel frames. The decapsulation and processing of a TRILL OAM Channel frame MAY, if provided for by the OAM protocol involved, result in the sending of one or more native TRILL OAM frames to one or more end stations.

6. Allocations Considerations

The following subsections give IANA and IEEE Registration Authority Considerations.

6.1 IANA Considerations

In this document, the allocation procedures "Standards Action", "IETF Review", "RFC Publication", and "Private Use" are as specified in [RFC5226].

IANA is requested to allocate a previously unassigned TRILL Nickname as follows:

```
      Any-RBridge           TBD (0xFFCO suggested)
```

IANA is requested to allocate two previously unassigned TRILL Multicast address as follows:

```
      All-OAM-RBridges      TBD (01-80-C2-00-00-43 suggested)
      TRILL-End-Stations    TBD (01-80-C2-00-00-44 suggested)
```

IANA is requested to allocate a previously unassigned TRILL critical ingress-to-egress extended flag bit as follows:

```
      TBD                   OAM-Flag
```

IANA is request to allocate a previously unassigned TRILL non-critical hop-by-hop extended flag bit as follows:

```
      TBD                   Alert
```

IANA is requested to create an additional sub-registry in the TRILL Parameter Registry for TRILL OAM Protocols, with initial contents as follows:

Protocol	Use
0x000	Reserved
0x001	OAM Channel Error
0x002-0x0FF	Available for allocation (1)
0x100-0xFF7	Available for allocation (2)
0xFF8-0xFFE	Private Use
0xFFF	Reserved

(1) TRILL OAM protocol code points from 0x002 to 0x0FF require a Standards Action for allocation.

(2) TRILL OAM protocol code points from 0x100 to 0xFF7 require RFC Publication to allocate a single value or IETF Review to allocate multiple values.

IANA is requested to create an additional sub-registry in the TRILL Parameter Registry for TRILL OAM Header Flags with initial contents as follows:

Flag Bit	Mnemonic	Allocation
0	SL	Silent
1	MH	Multi-hop
2-11	-	Available for allocation

Allocation of a TRILL OAM Header Flag is based on Standards Action [RFC5226].

IANA is requested to create an additional sub-registry in the TRILL Parameter Registry for TRILL OAM Channel error codes with initial contents as listed in Section 4.2 above and with available values allocated by Standards Action.

6.2 IEEE Registration Authority Considerations

The Ethertype TBD has been is assigned by the IEEE Registration Authority for TRILL-OAM.

7. Security Considerations

See [RFCtrill] for general RBridge Security Considerations.

-- More TBD --

8. References

The following sections list normative and informative references for this document.

8.1 Normative References

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

[RFC5226] - Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5880] - D. Katz, D. Ward, "Bidirectional Forwarding Detection (BFD)", June 2010.

[RFC5882] - D. Katz, D. Ward, "Generic Application of Bidirectional Forwarding Detection (BFD)", June 2010.

[RFCtrill] - R. Perlman, D. Eastlake, D. Dutt, S. Gai, and A. Ghanwani, "RBridges: Base Protocol Specification", draft-ietf-trill-rbridge-protocol-16.txt, in RFC Editor’s queue.

[RFCadj] - Eastlake, D., R. Perlman, A. Ghanwani, D. Dutt, V. Manral, "RBridges: Adjacency", draft-ietf-trill-adj, work in progress.

[TRILLopt] - D. Eastlake, A. Ghanwani, V. Manral, C. Bestler, "RBridges: TRILL Header Options", draft-ietf-trill-rbridge-options, work in progress.

8.2 Informative References

[RFC792] - Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[RFC5082] - Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007

[RFC5586] - Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, June 2009.

Authors' Addresses

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

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

Vishwas Manral
IP Infusion
1188 E. Arques Ave.
Sunnyvale, CA 94089 USA

Tel: +1-408-400-1900
EMail: vishwas@ipinfusion.com

Dave Ward
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089-1206 USA

Phone: +1-408-745-2000
EMail: dward@juniper.net

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

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

Sam Aldrin
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050 USA

Phone:
Email: sam.aldrin@huawei.com

Copyright, Disclaimer, and Additional IPR Provisions

TRILL H. Zhai
Internet-Draft F. Hu
Intended status: Standards Track ZTE Corporation
Expires: August 29, 2011 Feb 25, 2011

Extending the Virtual Router Redundancy Protocol for TRILL campus

draft-hu-trill-rbridge-vrrp-00.txt

Abstract

TRILL can be implemented in data center, which is request high reliability and stable, but if RBridge breaks down, the switch time is up to IS-IS topology convergence time. This is not satisfied to the data center service. VRRP provides a redundancy mechanism to avoid single point of failure and fast switching over. This draft proposes to extend VRRP protocol to TRILL in data center.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

TRILL (transparent Interconnection of Lots of Links) is a new technology merging the advantages of layer two and layer three technology[RFCtrill], and is designed to replace STP(Spanning Tree Protocol). The routing protocol IS-IS is used as control plane protocol to discover the topology and advertise link state. When the topology changes, IS-IS LSPs flood the link state to other adjacency. The topology convergence time is about dozens of seconds.

As TRILL deploys in many data centers, it’s necessary to interconnect the different data centers. The interconnection scenario is as figure 1. BRB (Border RBridge) is the border of data center, and all the data cross data center will get through BRB. If BRB is down, the cross data center communication will get down. BRB becomes the bottleneck of data center and is very easy to create a single point of failure. The solution is to provide redundant equipment to backup BRB. If BRB is broken down, the backup BRB can replace it. But the switching time is dependent the topology convergence time.However,it is request very high reliability in data center for providing video data application.

This draft propose to apply VRRP for ensure switching speed. The VRRP mechanism can implement the millisecond switching time to ensure video data [VRRPv3]. The BRB and backup BRB are configured as a VRRP group with the same virtual system ID and virtual nickname. The master BRB of the group floods the virtual nickname to adjacency.If the Master becomes unavailable then the highest priority Backup will be elected as Master after a short delay, providing a controlled transition of the virtual RBridge responsibility with minimal service interruption, and the master elected floods LSPs and data forwarding in TRILL campus, and the content of LSPs and the IS-IS link state topology doesn’t change.

Data Center Interconnection

```
+------------------------+            +-------------------------+
|                        |            |                         |
|           +-------+    |------------|    +-------+            |
| +-----+   |       |    |            |    |       |    +-----+ |
| | BR1|----+ BRB1 +----| IP Cloud   |----+ BRB2 +----|BR2 | |
| +-----+   |       |    |            |    |       |    +-----+ |
|           +-------+    |------------|    +-------+            |
|   Data Center 1        |            |     Data Center 2       |
+------------------------+            +-------------------------+
```

Figure 1

2. Terminology

Border RBridge: Abbr. BRB, a device locates the border of TRILL campus and runs TRILL protocol, BRB is used to communicate with other TRILL campus

VRRP RBridge: an RBridge running the Virtual Router Redundancy Protocol. It may participate in one or more VRRP groups.

Virtual RBridge: An abstract object managed by VRRP that acts as a default RBridge for devices on a shared LAN. It consists of a Virtual System Identifier and a set of associated nickname (s) across a common LAN. A VRRP RBridge may backup one or more virtual RBridges.

Nickname OwnerGBPoThe VRRP RBridge that has the virtual RBridge’s nickname as one of its nickname addresses. This is the RBridge that, when up, will respond to packets addressed to one of these nickname addresses for ICMP pings, TCP connections, etc.

Virtual RBridge masterGBPoThe VRRP RBridge that is assuming the responsibility of forwarding packets sent to the nickname associated with the virtual RBridge, and answering ARP requests for these nickname. Note that if the nickname owner is available, then it will always become the Master.

Virtual RBridge backupGBPoThe set of VRRP RBridge available to assume forwarding responsibility for a virtual RBridge should the current Master fail.

3. Application Scenario

The following figure shows a typical network with two VRRP BRBs implementing one virtual RBridge. One BRB is the virtual RBridge master, and the other BRB is virtual RBridge backup. BRB1 is assigned nickname owner of nickname A, and RBR2 is assigned nickname owner of nickname B. A virtual RBridge is then defined by associating a virtual nickname, which can be one of the nicknames of RBR1 and RBR2, or a different nickname from nickname A and nickname B. if virtual nickname is the nickname RB1, RBR1 is the nickname owner, then RBR1 is the virtual RBridge master automatically. Otherwise, the virtual RBridge master will be elected from RB1 and RB2 according to the priority. VRRP protocol manages virtual RBridge fail over to a backup RBridge. The master RBridge floods the IS-IS LSPs and data forwarding according to virtual system id and nickname(s) in TRILL campus.

```
               +-------------+      +-------------+
               |   BRB1      |      |   BRB2      |
               |(MRB VRBID=1)|      |(BRB VRBID=1)|
               |NICKNAME A   |      |NICKNAME B   |
               +------+------+      +----+--------+
                      |     VRID=1       |
                      |                  |
         NICKNAME A   |                  |
        -------+------+-----+------------+---+---------+----------
               |            |                |         |
               |            |                |         |
               |            |                |         |
            +--+--+      +--+--+          +--+--+   +--+--+
            | RB1 |      | RB2 |          | RB3 |   | RB4 |
            +-----+      +-----+          +--+--+   +--+--+

        Legend:
                 ---+---+---+--  =  Ethernet
                             BRB =  Border RBridge
                             RB  =  RBidge
                             MRB =  Master RBridge
                             BRB =  Backup RBridge
```

Figure 2

4. TRILL VRRP Frames

By multicasting periodically a TRILL VRRP frame, a master RBridge announces its existence and functionality to the backup RBridge(s) in a VRRP group. If none TRILL VRRP frame is received in a certain time, backup RBridge(s) will consider the master unavailable and trigger a new master RBridge election process.

A TRILL VRRP frame on an 802.3 link is structured as figure 3. All such frames are Ethertype encoded. The RBridge port out which such a frame is sent will strip the outer VLAN tag if configured to do so.

VRRP Frame Structure

```
 Outer Ethernet Header:
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                  TRILL-VRRP Multicast Address                   |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |      TRILL-VRRP continued     |  Source RBridge MAC Address     |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |           Source RBridge MAC Address continued                 |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |    Ethertype = C-Tag [802.1Q] | Outer.VLAN Tag Information     |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |    L2-TRILL-VRRP Ethertype    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

  VRRP for TRILL Payload:
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                         TRILL VRRP Payload                      |

  Frame Check Sequence:
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | FCS (Frame Check Sequence)                                      |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3

4.1. TRILL-VRRP Multicast Address

The TRILL-VRRP multicast address is an IP-derived multicast MAC address. The IP address is:

224.0.0.18

The IP-derived multicast address is a link local scope multicast address. RBridges MUST NOT forwards a frame with this destination address to another link.

4.2. Source RBridge MAC Address

It is a MAC address of RBridge port out which this TRILL VRRP frame is sent

4.3. L2-TRILL-VRRP Ethertype

It is used to indicate that the payload in the frame is a TRILL VRRP packet

4.4. Frame Check Sequence (FCS)

Each Ethernet frame has a single Frame Check Sequence (FCS) that is computed to cover the entire frame, for detecting frame corruption due to bit errors on a link. Thus, when a frame is encapsulated, the original FCS is not included but is discarded. Any received frame for which the FCS check fails SHOULD be discarded (this may not be possible in the case of cut through forwarding).

Although the FCS is normally calculated just before transmission, it is desirable, when practical, for an FCS to accompany a frame within an RBridge after receipt.

5. TRILL VRRP Payload Format

The format of TRILL VRRP payload is structured as figure 4.

VRRP Payload Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Type  | Virtual RB ID |   Priority    |Count Nicknames|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|(rsvd) |     Max Adver Int     |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Virtual System ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Virtual System ID Continued |          Nickname (1)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Nickname (2)       |         Nickname (3)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Nickname (n-1)       |           Nickname (n)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4

5.1. Version

The version field specifies the TRILL VRRP protocol version of this packet. This document defines version 1.

5.2. Type

The type field specifies the type of this TRILL VRRP packet. The only packet type defined in this version of the protocol is:

1 ADVERTISEMENT

A packet with unknown type MUST be discarded.

5.3. Virtual RB ID

The Virtual RBridge Identifier (VRBID) field identifies the virtual RBridge this packet is reporting status for. It is a configurable item in the range 1-255 (decimal). There is no default.

5.4. Priority

The priority field specifies the sending TRILL VRRP RBridge's priority for the virtual RBridge. Higher values equal higher priority. This field is an 8-bit unsigned integer field.

The priority value for the TRILL VRRP RBridge that owns the nicknames associated with the virtual nickname MUST be 255 (decimal).

TRILL VRRP RBridges backing up a virtual RBridge MUST use priority values between 1-254 (decimal) and the default priority value is 100(decimal).

The priority value zero (0) has special meaning, indicating that the current Master has stopped participating in TRILL VRRP. This is used to trigger backup RBridges to quickly transition to Master without having to wait for the current Master to time out.

5.5. Count Nicknames

The number of nicknames contained in this TRILL VRRP advertisement.

5.6. Rsvd

This field MUST be set to zero on transmission and ignored on reception.

5.7. Maximum Advertisement Interval (Max Adver Int)

The Maximum Advertisement Interval is a 12-bit field that indicates the time interval (in centiseconds) between ADVERTISEMENTS. The default is 100 centiseconds (1 second).

5.8. Checksum

The checksum field is used to detect data corruption in the TRILL VRRP message.

The checksum is the 16-bit one’s complement of the one’s complement sum of the entire TRILL VRRP message starting with the version field. For computing the checksum, the checksum field is set to zero. See RFC1071 for more detail [CKSM].

5.9. Virtual System ID

The virtual system id is a 48-bit field that indicates the system id of the virtual RBridge this packet is reporting status for.

All the RBridges in a virtual RBridge MUST be configured with the same virtual system id. When a TRILL VRRP packet with different virtual system id from local virtual system id is received, the packet MUST be discarded. This field is used for troubleshooting misconfigured RBridges.

5.10. Nickname(s)

One or more nicknames are associated with the virtual RBridge. The number of nicknames included is specified in the "Count Nicknames" field. These fields are used for troubleshooting misconfigured RBridges.

6. VRRP Protocol State Machine

The VRRP protocol state machine is not change. There are three states: Initialize, backup and master. Initialize state is to wait for a startup event; backup state is to monitor the availability and state of the master RBridge.

The master BRB election is according to the priority value. When the RBridge is elected as virtual RBridge master, it floods LSP with virtual nickname to its’ adjacencies. If the RBridge is the nickname owner, it’s the virtual nickname master automatically, and floods LSPs with owner nickname. Backup RBridge monitors and receives the VRRP packet from master. If backup RBridge has already enabled IS-IS protocol, it should flood LSP to withdraw its nickname LSA. Otherwise backup RBridge shouldn’t flood LSP to its neighbors. Backup RBridge exchanges hello packet with its neighbor, and receives LSP from its adjacencies except master RBridgeGBP[not]but never advertises local LSA, which is advertised by master RBridge.

7. IS-IS Adjacency

Master RBridge should setup and maintain all the adjacencies with other RBridges except backup RBridge. Backup RBridge receives the other RBridges hello packets and IS-IS packets (such as LSP, CSNP, PSNP) besides master RBridge, but should not send any hello and IS-IS packets (LSP, CSNP, PSNP) to other RBridges. The backup RBridge can be detect, 2-way, and report states [TrillAdj].

8. Security Considerations

9. Acknowledgements

The authors would like to gratefully acknowledge many people who have contributed discussion and ideas to the making of this proposal. They include Lizhong Jin, Mingjiang Cheng,Min Xiao, Bo Wu, Xiefeng Gong, Jingjing Zhao, Erchun Lv,etc.

10. References

10.1. Normative references

[RFC1071] Braden, R., Borman, D., Partridge, C., and W. Plummer, "Computing the Internet checksum", RFC 1071, September 1988.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[RFC3768] Hinden, R., "Virtual Router Redundancy Protocol (VRRP)", RFC 3768, April 2004.

[RFC5798] Nadas, S., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, March 2010.

[RFCtrill]
Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "RBridges: Base Protocol Specification", draft-ietf- trill-rbridge-protocol-16.txt, in RFC Editor’s queue, Mar 2010.

[TrillAdj]
Eastlake, D., Perlman, R., Ghanwani, A., Dutt, D., and V. Manral, "RBridges: Adjacency", draft-ietf-trill-adj-02.txt, work in process, Feb 2011.

10.2. Informative References

Authors' Addresses

Hongjun Zhai
ZTE Corporation
68 Zijinghua Road
Nanjing 200012
China

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

Fangwei Hu
ZTE Corporation
889 Bibo Road
Shanghai 201203
China

Phone: +86-21-68896273
Email: hu.fangwei@zte.com.cn

TRILL Working Group
Internet-Draft
Intended status: Proposed Standard
Expires: September 3, 2011
A. Rijhsinghani
Hewlett-Packard
K. Zebrose
H.W. Embedded
March 2, 2011

Definitions of Managed Objects for RBridges
draft-ietf-trill-rbridge-mib-02.txt

Status of This Document

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols. In particular it defines objects for managing RBridges, which are devices that implement the TRILL protocol.

Table of Contents

1. Introduction

This document describes a model for managing RBridges as defined in [RBridge]. RBridges provide optimal pair-wise forwarding without configuration using IS-IS routing and encapsulation of traffic. RBridges are compatible with previous IEEE 802.1 customer bridges as well as IPv4 and IPv6 routers and end nodes. They are as invisible to current IP routers as bridges are and, like routers, they terminate the bridge spanning tree protocol. In creating an RBridge management model the device is viewed primarily as a customer bridge. For a discussion of the problem addressed by TRILL see [RFC5556].

RBridges support features specified for transparent bridges in IEEE 802.1, and the corresponding MIBs are used to manage those features. For IS-IS purposes, the corresponding MIB is used to manage the protocol. This MIB specifies those objects which are TRILL-specific and hence not available in other MIBs.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [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 mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, which consists of [RFC2578], [RFC2579] and [RFC2580].

3. Overview

The RBridge MIB is intended as an overall framework for managing RBridges. Where possible the MIB references existing MIB definitions in order to maximize reuse. This results in a considerable emphasis on the relationship with other MIB documents.

Starting with the physical interfaces, there are requirements for certain elements of the IF-MIB to be implemented. These elements are required in order to connect the per-port parameters to higher level functions of the physical device.

Transparent bridging, VLANs, Traffic classes and Multicast Filtering are supported by the TRILL protocol, and the corresponding management is expected to conform to the BRIDGE-MIB [RFC4188], P-BRIDGE-MIB and Q-BRIDGE-MIB [RFC4363] modules.

The IS-IS routing protocol is used in order to determine the optimum pair-wise forwarding path. This protocol is managed using the IS-IS MIB defined in [RFC4444]. Since the TRILL protocol specifies use of a single level and a fixed area address of zero, some MIB objects are not applicable. Some IS-IS MIB objects are used in the TRILL protocol.

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 are arranged into subtrees. Each subtree is organized as a set of related objects. The various subtrees are shown below. These are supplemented with required elements of the IF-MIB, ISIS-MIB, BRIDGE-MIB, P-BRIDGE-MIB and Q-BRIDGE-MIB.

5.1 Textual Conventions

Textual conventions are defined to represent object types relevant to TRILL.

5.2 The rbridgeBase Subtree

This subtree contains system and port specific objects applicable to all RBridges.

5.3 The rbridgeFdb Subtree

This subtree contains objects applicable to the Forwarding database used by the RBridge in making packet forwarding decisions. Because it contains additional information used by the TRILL protocol not applicable to 802.1D/Q bridges, it is a superset of the corresponding subtrees defined in the BRIDGE-MIB and Q-BRIDGE-MIB.

5.4 The rbridgeVlan subtree

This subtree describes objects applicable to VLANs configured on the RBridge.

5.5 The rbridgeEsadi subtree

This subtree describes objects relevant to RBridges that support the optional ESADI protocol.

5.6 The rbridgeCounters subtree

This subtree contains statistics maintained by RBridges that can aid in monitoring and troubleshooting networks connected by them.

5.7 The rbridgeSnooping subtree

This subtree describes objects applicable to RBridges capable of snooping IPv4 and/or IPv6 Multicast control frames and pruning IP multicast traffic based on detection of IP multicast routers and listeners.

5.8 The rbridgeDtree subtree

This subtree contains objects relevant to Distribution Trees computed by RBridges for the forwarding of multi-destination frames.

5.9 The rbridgeTrill subtree

This subtree contains objects applicable to the TRILL IS-IS protocol, beyond what is available in ISIS-MIB.

5.10 The Notifications Subtree

The defined notifications are focused on the TRILL protocol functionality. Notifications are defined for changes in the Designated RBridge status and the topology. TBD for this section is what notifications are required from imported MIBs and how can the TRILL notifications be throttled.

6. Relationship to Other MIB Modules

The IF-MIB, BRIDGE-MIB, P-BRIDGE-MIB, Q-BRIDGE-MIB, and ISIS-MIB all contain objects relevant to the RBridge MIB. Management objects contained in these modules are not duplicated here, to reduce overlap to the extent possible.

6.1 Relationship to IF-MIB

The port identification elements MUST be implemented in order to allow them to be cross referenced. The Interface MIB [RFC2863] requires that any MIB module which is an adjunct of the Interface MIB clarify specific areas within the Interface MIB. These areas were intentionally left vague in the Interface MIB to avoid over-constraining the MIB, thereby precluding management of certain media-types. Section 4 of [RFC2863] enumerates several areas which a media-specific MIB must clarify. The implementor is referred to

[RFC2863] in order to understand the general intent of these areas.

6.2 Relationship to BRIDGE-MIB

The following subtrees in the BRIDGE-MIB [RFC4188] contain information relevant to RBridges when the corresponding functionality is implemented. This functionality is also contained in IEEE8021-BRIDGE-MIB.

- o dot1dBase
- o dot1dTp
- o dot1dStatic

6.3 Relationship to P-BRIDGE-MIB

The following subtrees in the P-BRIDGE-MIB [RFC4363] contain information relevant to RBridges when the corresponding functionality is implemented. This functionality is also contained in IEEE8021-BRIDGE-MIB.

- o dot1dExtBase
- o dot1dPriority
- o dot1dGarp
- o dot1dGmrp
- o dot1dTpHCPortTable
- o dot1dTpPortOverflowTable

6.4 Relationship to Q-BRIDGE-MIB

The following groups in the Q-BRIDGE-MIB [RFC4363] contain information relevant to RBridges when the corresponding functionality is implemented. This functionality is also contained in IEEE8021-Q-BRIDGE-MIB.

- o dot1qBase
- o dot1qTp
- o dot1qStatic
- o dot1qVlan
- o dot1vProtocol

6.5 Relationship to IS-IS MIB

The Management Information Base for Intermediate System to Intermediate System (IS-IS)[RFC4444] defines a MIB for the IS-IS Routing protocol when it is used to construct routing tables for IP networks. While most of these objects are directly applicable to the TRILL layer 2 implementations there are some modifications detailed below.

System-Wide Attributes

isisSystem -

This table contains information specific to a single instance of the IS-IS protocol. The TRILL IS-IS implementation follows the IS-IS MIB except for the following changes:

isisLevelType MUST read level 1

The TRILL IS-IS implementation does not include Level 2.

isisSysProtSupport MUST read zero

The IP protocols detailed in the IS-IS MIB are not applicable.

isisSysL2toL2Leaking MUST read FALSE

The TRILL IS-IS implementation does not include Level 2.

isisManAreaAddr -

This subtree is not implemented in TRILL IS-IS. TRILL IS-IS uses a single fixed area address of zero.

isisAreaAddr -

This subtree is not implemented in TRILL IS-IS. TRILL IS-IS uses a single fixed area address of zero.

isisSummAddr -

This subtree is not implemented in TRILL IS-IS. In IS-IS this table holds summary addresses configured for each Level 2 instance of the IS-IS protocol running on a router. TRILL does not implement Level 2.

isisRedistributeAddr -

This subtree is not implemented in TRILL IS-IS. In IS-IS this table is used to implement Level2 to Level1 address leaking. TRILL does not implement Level 2.

isisRouter -

This table is implemented. This table holds the System ID for Intermediate Systems in the campus.

isisSysLevel -

This table is implemented. This table contains information specific to a domain (Level 2) or an area (Level 1) of the IS-IS protocol. In the case of TRILL IS-IS there is only one entry in the table for Level 1 area zero.

isisNextCircIndex -

This scalar is implemented. This scalar is used to provide a unique circuit index.

Circuit-specific Attributes

isisCirc -

This table is implemented, with the following modification. This table contains information specific to a point-to-point or a broadcast interface in the system.

isisCircLevelType MUST read level1

isisCircLevelIndex MUST read level1

Counters

isisSystemCounter -

This table is implemented. Counters in the System table, such as number of times we have wrapped a sequence counter on one of our Link State PDUs.

isisCircuitCounter -

This table is implemented. Counters of events particular to a circuit, such as PDUs with an illegal value of the System ID field length.

isisPacketCounter -

This table is implemented. Counts of IS-IS Protocol PDUs broken down into packet type.

Attributes associated with an Adjacency

isisISAdj -

This table is implemented. This table contains information about adjacencies to RBridges maintained by the protocol. Entries in this table cannot be created by management action: they are established through the Hello protocol.

isisISAdjAreaAddr -

This table is not implemented. This table contains the set of Area Addresses of neighboring Intermediate Systems, as reported in IIH PDUs. Since all area addresses are zero there is no need for a table.

isisISAdjIPAddr -

This table is not implemented. This table contains the set of IP Addresses of neighboring Intermediate Systems, as reported in received IIH PDUs. The table has been replaced by addition of the RBridgeISAdjMACAddr in the RBridge subtree.

isisISAdjProtSupp -

This table is not implemented. This table contains the set of protocols supported by neighboring Intermediate Systems, as reported in received IIH PDUs.

Attributes Associated with Addresses

isisRA -

This table is implemented. The Reachable Address Table.

Normally each entry defines a configured Reachable Address to an NSAP or Address Prefix. In the case of an RBridge the unique isisRAIndex should be defined as type MacAddress rather than an Unsigned32.

isisIPRA -

This table is not implemented. The IP Reachable Address Table.

This table contains information about an IP reachable address manually configured on this system or learned from another protocol.

Attributes Associated with Link State PDU Table

isisLSPSummaryTable -

This table is implemented. The Link State PDU Summary Table.

This table contains information contained in the headers of Link State PDUs stored by the system.

isisLSPTLVTable -

This table is implemented. The Link State PDU TLV Table.

This table holds the sequence of TLVs that make up an LSP fragment.

Attributes Associated with a Notification

isisNotification

This table is implemented. This table defines attributes that will be included when reporting IS-IS notifications.

6.6 MIB modules required for IMPORTS

The following MIB module IMPORTS objects from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], and IF-MIB [RFC2863].

7. Definition of the RBridge MIB

```
 RBRIDGE-MIB DEFINITIONS ::= BEGIN

   -- ---------------------------------------------------------------- --
   -- MIB for RBRIDGE devices
   -- ---------------------------------------------------------------- --
   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
       Counter32, Integer32, mib-2
           FROM SNMPv2-SMI            -- RFC2578
       TEXTUAL-CONVENTION, TruthValue, MacAddress
           FROM SNMPv2-TC             -- RFC2579
       MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
           FROM SNMPv2-CONF
       VlanId, PortList, dot1qFdbId, dot1qVlanIndex
           FROM Q-BRIDGE-MIB
       InetAddress, InetAddressType
           FROM INET-ADDRESS-MIB
```

```
        BridgeId
            FROM BRIDGE-MIB
        InterfaceIndex
            FROM IF-MIB
        ;

   rbridgeMIB MODULE-IDENTITY
   LAST-UPDATED "201003010000Z"
   ORGANIZATION "IETF TRILL Working Group"
   CONTACT-INFO
        "http://www.ietf.org/dyn/wg/charter/trill-charter.html
        Email: rbridge@postel.org

               Anil Rijhsinghani
               Hewlett-Packard
          Tel: +1 508 323 1251
        Email: anil@charter.net

               Kate Zebrose
               H.W. Embedded
          Tel: +1 617 840 9673
        Email: kate.zebrose@alum.mit.edu"

        DESCRIPTION
            "The RBridge MIB module for managing devices that support
            the TRILL protocol."

   REVISION     "201103010000Z"
   DESCRIPTION
        "Initial version, published as RFC yyyy"
-- RFC Ed.: replace yyyy with actual RFC number & remove this note

        ::= { mib-2 xxx }
-- RFC Ed.: replace xxx with  IANA-assigned number & remove this note

   -- ------------------------------------------------------------- --
   -- subtrees in the RBridge MIB
   -- ------------------------------------------------------------- --

   rbridgeNotifications  OBJECT IDENTIFIER ::= { rbridgeMIB 0 }
   rbridgeObjects        OBJECT IDENTIFIER ::= { rbridgeMIB 1 }
   rbridgeConformance    OBJECT IDENTIFIER ::= { rbridgeMIB 2 }

   rbridgeBase           OBJECT IDENTIFIER ::= { rbridgeObjects 1 }
   rbridgeFdb            OBJECT IDENTIFIER ::= { rbridgeObjects 2 }
   rbridgeVlan           OBJECT IDENTIFIER ::= { rbridgeObjects 3 }
   rbridgeEsadi          OBJECT IDENTIFIER ::= { rbridgeObjects 4 }
   rbridgeCounter        OBJECT IDENTIFIER ::= { rbridgeObjects 5 }
```

```
   rbridgeSnooping       OBJECT IDENTIFIER ::= { rbridgeObjects 6 }
   rbridgeDtree          OBJECT IDENTIFIER ::= { rbridgeObjects 7 }
   rbridgeTrill          OBJECT IDENTIFIER ::= { rbridgeObjects 8 }

   -- ------------------------------------------------------------ --
   -- type definitions
   -- ------------------------------------------------------------ --

   RbridgeAddress ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "1x:"
       STATUS current
       DESCRIPTION
           "The MAC address used by an RBridge port. This may match the
           RBridge ISIS SystemID."
   SYNTAX OCTET STRING (SIZE (6))


   RbridgeNickname ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "d"
       STATUS current
       DESCRIPTION
           "The 16-bit identifier used in TRILL as an
           abbreviation for the RBridge's 48-bit IS-IS System ID.
           The value 0 means a nickname is not specified, the values
           0xffco through 0xfffe are reserved for future allocation,
           and the value 0xffff is permanently reserved."
   SYNTAX Integer32 (0..65471)

   --
   -- the rbridgeBase subtree
   --
   -- Implementation of the rbridgeBase subtree is mandatory for all
   -- RBridges.
   --

   rbridgeBaseTrillVersion OBJECT-TYPE
       SYNTAX      Integer32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The maximum TRILL version number that this Rbridge
           supports."
       REFERENCE
           "RBridge section 4.6"
       ::= { rbridgeBase 1 }

   rbridgeBaseNumPorts OBJECT-TYPE
      SYNTAX      Integer32
```

```
      UNITS       "ports"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The number of ports controlled by this RBridge."
     REFERENCE
          "RBridge section 2.6.1"
      ::= { rbridgeBase 2 }

   rbridgeBaseForwardDelay OBJECT-TYPE
       SYNTAX      Integer32
       UNITS       "seconds"
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
           "Modified aging time for address entries after an appointed
           forwarder change. The default value is 15."
       REFERENCE
            "RBridge section 4.8.2"
      ::= { rbridgeBase 3 }

   rbridgeBaseUniMultipathEnable OBJECT-TYPE
       SYNTAX      INTEGER {
                       enabled(1),
                       disabled(2)
                   }
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
           "The enabled/disabled status of unicast TRILL
           multipathing."
       REFERENCE
            "RBridge Appendix C"
      ::= { rbridgeBase 4 }

   rbridgeBaseMultiMultipathEnable OBJECT-TYPE
       SYNTAX      INTEGER {
                       enabled(1),
                       disabled(2)
                   }
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
           "The enabled/disabled status of multidestination TRILL
           multipathing."
       REFERENCE
            "RBridge Appendix C"
      ::= { rbridgeBase 5 }
```

```
   rbridgeBaseNicknameNumber OBJECT-TYPE
       SYNTAX      Integer32 (0..255)
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
           "The number of nicknames this RBridge should have.
           Default value is 1."
      ::= { rbridgeBase 6 }

   rbridgeBaseAcceptEncapNonadj OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
           "Accept TRILL-encapsulated frames from a neighbor with which
           this RBridge does not have an IS-IS adjacency. The default
           is false."
       REFERENCE
            "RBridge section 4.6.2"
      ::= { rbridgeBase 7 }

   -- ------------------------------------------------------------ --
   -- The RBridge Base Nickname Table
   -- ------------------------------------------------------------ --

   rbridgeBaseNicknameTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF RbridgeBaseNicknameEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
       "A table that contains information about nicknames
       associated with this RBridge."
      REFERENCE
           "RBridge section 3.7"
      ::= { rbridgeBase 8 }

   rbridgeBaseNicknameEntry OBJECT-TYPE
       SYNTAX      RbridgeBaseNicknameEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A list of information for each nickname of the RBridge."
       REFERENCE
           "RBridge section 3.7"
       INDEX  { rbridgeBaseNicknameName }
       ::= { rbridgeBaseNicknameTable 1 }

   RbridgeBaseNicknameEntry ::=
```

```
        SEQUENCE {
            rbridgeBaseNicknameName
                RbridgeNickname,
            rbridgeBaseNicknamePriority
                Integer32,
            rbridgeBaseNicknameDtrPriority
                Integer32,
            rbridgeBaseNicknameStatus
                INTEGER
        }

    rbridgeBaseNicknameName OBJECT-TYPE
        SYNTAX      RbridgeNickname
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Nicknames are 16-bit quantities that act as
             abbreviations for RBridge's 48-bit IS-IS System ID to
             achieve a more compact encoding."
        REFERENCE
            "RBridge section 3.7"
        ::= { rbridgeBaseNicknameEntry 1 }

    rbridgeBaseNicknamePriority OBJECT-TYPE
        SYNTAX      Integer32 (0..255)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "This RBridge's priority to hold this nickname. When
            the nickname is configured, the default value of
            this object is 192."
        REFERENCE
            "RBridge section 3.7"
        DEFVAL      { 192 }
        ::= { rbridgeBaseNicknameEntry 2 }

    rbridgeBaseNicknameDtrPriority OBJECT-TYPE
        SYNTAX      Integer32 (1..65535)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The Distribution tree root priority for this nickname.
            The default value of this object is 32768."
        REFERENCE
            "RBridge section 4.5"
        DEFVAL      { 32768 }
        ::= { rbridgeBaseNicknameEntry 3 }
```

```
   rbridgeBaseNicknameStatus OBJECT-TYPE
       SYNTAX      INTEGER {
                       static(1),
                       dynamic(2),
                       invalid(3)
                   }
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "This object indicates the status of the entry. The
           default value is static(1).
               static(1) - this entry has been configured and
                   will remain after the next reset of the RBridge.
               dynamic(2) - this entry has been acquired by the
                   RBridge nickname acquisition protocol.
               invalid(3) - writing this value to the object removes
                   the corresponding entry."
       REFERENCE
           "RBridge section 3.7"
       DEFVAL      { static }
       ::= { rbridgeBaseNicknameEntry 4 }


   -- ---------------------------------------------------------- --
   -- The RBridge Port Table
   -- ---------------------------------------------------------- --

   rbridgeBasePortTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF RBridgeBasePortEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A table that contains generic information about every
           port that is associated with this RBridge."
       REFERENCE
           "RBridge section 5.2"
       ::= { rbridgeBase 9 }

   rbridgeBasePortEntry OBJECT-TYPE
       SYNTAX      RBridgeBasePortEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A list of information for each port of the bridge."
       REFERENCE
           "RBridge section 5.2"
       INDEX  { rbridgeBasePort }
       ::= { rbridgeBasePortTable 1 }
```

```
   RBridgeBasePortEntry ::=
       SEQUENCE {
           rbridgeBasePort
               Integer32,
           rbridgeBasePortIfIndex
               InterfaceIndex,
           rbridgeBasePortDisable
               TruthValue,
           rbridgeBasePortTrunkPort
               TruthValue,
           rbridgeBasePortAccessPort
               TruthValue,
           rbridgeBasePortP2pHellos
               TruthValue,
           rbridgeBasePortState
               INTEGER,
           rbridgeBasePortInhibitionTime
               Integer32,
           rbridgeBasePortDisableLearning
               TruthValue,
           rbridgeBasePortDesiredDesigVlan
               VlanId,
           rbridgeBasePortDesigVlan
               VlanId,
           rbridgeBasePortStpRoot
               BridgeId,
           rbridgeBasePortStpRootChanges
               Counter32,
           rbridgeBasePortStpWiringCloset
               BridgeId
   }

   rbridgeBasePort OBJECT-TYPE
       SYNTAX      Integer32 (1..65535)
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "The port number of the port for which this entry
           contains RBridge management information."
       REFERENCE
           "RBridge section 5.2"
       ::= { rbridgeBasePortEntry 1 }

   rbridgeBasePortIfIndex OBJECT-TYPE
       SYNTAX      InterfaceIndex
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
```

```
            "The value of the instance of the ifIndex object,
            defined in IF-MIB, for the interface corresponding
            to this port."
        ::= { rbridgeBasePortEntry 2 }

   rbridgeBasePortDisable OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "Disable port bit. When this bit is set (true), all frames
            received or to be transmitted are discarded, with the
            possible exception of some layer 2 control frames that may
            be generated and transmitted or received and processed
            locally. Default value is false."
        REFERENCE
           "RBridge section 4.9.1"
        DEFVAL      { false }
        ::= { rbridgeBasePortEntry 3 }

   rbridgeBasePortTrunkPort OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "End station service disable (trunk port) bit. When this bit
            is set (true), all native frames received on the port and all
            native frames that would have been sent on the port are
            discarded. Default value is false."
        REFERENCE
           "RBridge clause 4.9.1"
        DEFVAL      { false }
        ::= { rbridgeBasePortEntry 4 }

   rbridgeBasePortAccessPort OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "TRILL traffic disable (access port) bit. If this bit is set,
            the goal is to avoid sending any TRILL frames, except
            TRILL-Hello frames, on the port since it is intended only for
            native end station traffic.  This ensures that the link is
            not on the shortest path for any destination. Default value
            is false."
        REFERENCE
           "RBridge clause 4.9.1"
        DEFVAL      { false }
```

```
       ::= { rbridgeBasePortEntry 5 }

   rbridgeBasePortP2pHellos OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "Use P2P Hellos bit. If this bit is set, Hellos sent on this
           port are IS-IS P2P Hellos, not the default TRILL-Hellos. In
           addition, the IS-IS P2P three-way handshake is used on P2P
           RBridge links. Default value is false."
       REFERENCE
           "RBridge clause 4.9.1"
       DEFVAL      { false }
       ::= { rbridgeBasePortEntry 6 }

   rbridgeBasePortState OBJECT-TYPE
       SYNTAX      INTEGER {
                          uninhibited(1),
                          portInhibited(2),
                          vlanInhibited(3),
                          disabled(4),
                          broken(5)
                      }
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The port's current state. If the entire port is
           inhibited, its state is portInhibited(2). If specific VLANs
           are inhibited, the state is vlanInhibited(3) and
           rbridgeVlanTable will tell which VLANs are inhibited.
           For ports that are disabled (see rbridgeBasePortDisable),
           this object will have a value of disabled(4). If the
           RBridge has detected a port that is malfunctioning, it will
           place that port into the broken(5) state."
      REFERENCE
          "RBridge section 4.2.4.3"
      ::= { rbridgeBasePortEntry 7 }

   rbridgeBasePortInhibitionTime OBJECT-TYPE
       SYNTAX      Integer32
       UNITS       "seconds"
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "Time in seconds that this RBridge will inhibit forwarding
           on this port after it observes a spanning tree root bridge
           change on a link, or receives conflicting VLAN forwarder
```

```
            information. The default value is 30."
        REFERENCE
             "RBridge section 4.2.4.3"
        DEFVAL      { 30 }
        ::= { rbridgeBasePortEntry 8 }

    rbridgeBasePortDisableLearning OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "Disable learning of MAC addresses seen on this port.
            The default is false."
        REFERENCE
             "RBridge section 4.8"
        DEFVAL      { false }
        ::= { rbridgeBasePortEntry 9 }

    rbridgeBasePortDesiredDesigVlan OBJECT-TYPE
        SYNTAX      VlanId
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The VLAN that a DRB will specify in its TRILL-Hellos as the
            VLAN to be used by all RBridges on the link for TRILL frames.
            This VLAN must be enabled on this port."
        REFERENCE
            "RBridge section 4.4.3"
        ::= { rbridgeBasePortEntry 10 }

    rbridgeBasePortDesigVlan OBJECT-TYPE
        SYNTAX      VlanId
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The VLAN being used on this link for TRILL frames."
        REFERENCE
            "RBridge section 4.4.3"
        ::= { rbridgeBasePortEntry 11 }

    rbridgeBasePortStpRoot OBJECT-TYPE
        SYNTAX      BridgeId
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The bridge identifier of the root of the spanning
            tree, as learned from a BPDU received on this port. For
            MSTP, this is the root bridge of the CIST. If no BPDU has
```

```
            been heard, the value returned is a string of zeros."
        REFERENCE
            "RBridge section 4.2.4.3"
        ::= { rbridgeBasePortEntry 12 }

    rbridgeBasePortStpRootChanges OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The number of times a change in the root bridge is seen from
            spanning tree BPDUs received on this port, indicating a
            change in bridged LAN topology. Each such change may cause
            the port to be inhibited for a period of time."
        REFERENCE
            "RBridge section 4.9.3.2"
        ::= { rbridgeBasePortEntry 13 }

    rbridgeBasePortStpWiringCloset OBJECT-TYPE
        SYNTAX      BridgeId
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The Bridge ID to be used as Spanning Tree root in BPDUs
            sent for the Wiring Closet topology solution described in
            [RBridge]. Note that the same value of this object must be
            set on all RBridge ports participating in this solution.
            The default value is all 0s. A non-zero value configured
            into this object indicates that this solution is in use."
        REFERENCE
            "RBridge section A.3.3"
        ::= { rbridgeBasePortEntry 14 }

    -- ----------------------------------------------------------------
    -- RBridge Forwarding Database
    -- ----------------------------------------------------------------

    rbridgeConfidenceNative OBJECT-TYPE
        SYNTAX      Integer32 (0..255)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The confidence level associated with MAC addresses
            learned from native frames. The default value is 32."
        REFERENCE
             "RBridge section 4.8.1"
       ::= { rbridgeFdb 1 }
```

```
   rbridgeConfidenceDecap OBJECT-TYPE
       SYNTAX      Integer32 (0..255)
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
           "The confidence level associated with inner MAC addresses
           learned after decapsulation of a TRILL data frame.
           The default value is 32."
       REFERENCE
            "RBridge Appendix section 4.8.1"
      ::= { rbridgeFdb 2 }

   rbridgeConfidenceStatic OBJECT-TYPE
       SYNTAX      Integer32 (0..255)
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
           "The confidence level associated with MAC addresses that
           are statically configured. The default value is 255."
       REFERENCE
            "RBridge section 4.8.2"
       DEFVAL      { 255 }
       ::= { rbridgeFdb 3 }



   -- ----------------------------------------------------------------
   -- Multiple Forwarding Databases for RBridges
   -- This allows for an instance per FdbId, defined in Bridge MIB.
   -- Each VLAN may have an independent Fdb, or multiple VLANs may
   -- share one.
   -- ----------------------------------------------------------------

   rbridgeUniFdbTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF RbridgeUniFdbEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A table that contains information about unicast entries
           for which the device has forwarding and/or filtering
           information.  This information is used by the
           transparent bridging function in determining how to
           propagate a received frame."
       REFERENCE
           "RBridge section 4.8"
       ::= { rbridgeFdb 4 }

   rbridgeUniFdbEntry OBJECT-TYPE
```

```
        SYNTAX      RbridgeUniFdbEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Information about a specific unicast MAC address for
            which the rbridge has some forwarding and/or filtering
            information."
        INDEX   { dot1qFdbId, rbridgeUniFdbAddr }
        ::= { rbridgeUniFdbTable 1 }

    RbridgeUniFdbEntry ::=
        SEQUENCE {
            rbridgeUniFdbAddr
                MacAddress,
            rbridgeUniFdbPort
                Integer32,
            rbridgeUniFdbNick
                RbridgeNickname,
            rbridgeUniFdbConfidence
                Integer32,
            rbridgeUniFdbStatus
                INTEGER
        }

    rbridgeUniFdbAddr OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "A unicast MAC address for which the device has
            forwarding information."
        ::= { rbridgeUniFdbEntry 1 }

    rbridgeUniFdbPort OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Either the value '0', or the port number of the port on
            which a frame having a source address equal to the value
            of the corresponding instance of rbridgeUniFdbAddress has
            been seen.  A value of '0' indicates that the port
            number has not been learned but that the device does have
            some information about this MAC address.
            Implementors are encouraged to assign the port value to
            this object whenever it is available, even for addresses
            for which the corresponding value of rbridgeUniFdbStatus is
            not learned(3)."
```

```
       ::= { rbridgeUniFdbEntry 2 }

   rbridgeUniFdbNick OBJECT-TYPE
       SYNTAX      RbridgeNickname
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The RBridge nickname which is placed in the Egress
           Nickname field of a TRILL frame sent to this
           rbridgeFdbAddress in this FdbId."
      REFERENCE
           "RBridge section 4.8.1"
       ::= { rbridgeUniFdbEntry 3 }

   rbridgeUniFdbConfidence OBJECT-TYPE
       SYNTAX      Integer32 (0..254)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The confidence level associated with this entry."
      REFERENCE
           "RBridge section 4.8.1"
       ::= { rbridgeUniFdbEntry 4 }

   rbridgeUniFdbStatus OBJECT-TYPE
       SYNTAX      INTEGER {
                       other(1),
                       invalid(2),
                       learned(3),
                       self(4),
                       mgmt(5),
                       esadi(6)
                   }
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The status of this entry.  The meanings of the values
           are:
               other(1) - none of the following.
               invalid(2) - this entry is no longer valid (e.g., it
                   was learned but has since aged out), but has not
                   yet been flushed from the table.
               learned(3) - the information in this entry was learned
                   and is being used.
               self(4) - the value of the corresponding instance of
                   rbridgeFdbAddress represents one of the device’s
                   addresses.  The corresponding instance of
                   rbridgeFdbPort indicates which of the device’s
```

```
                   ports has this address.
               mgmt(5) - the value of the corresponding instance of
                   rbridgeFdbAddress was configured by management.
               esadi(6) - the value of the corresponding instance of
                   rbridgeFdbAddress was learned from ESADI."
       ::= { rbridgeUniFdbEntry 5 }

   -- ----------------------------------------------------------------
   -- RBridge FIB
   -- ----------------------------------------------------------------

   rbridgeUniFibTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF RbridgeUniFibEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A table that contains information about nicknames
           known by the RBridge. If ECMP is implemented, there are
           as many entries for a nickname as ECMP paths available for
           it."
       ::= { rbridgeFdb 5 }

   rbridgeUniFibEntry OBJECT-TYPE
       SYNTAX      RbridgeUniFibEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A list of information about nicknames known by the RBridge.
           If ECMP is implemented, there are as many entries as ECMP
           paths available for a given nickname."
       INDEX   { rbridgeFibNickname, rbridgeFibPort }
       ::= { rbridgeUniFibTable 1 }

   RbridgeUniFibEntry ::=
       SEQUENCE {
           rbridgeFibNickname
               RbridgeNickname,
           rbridgeFibPort
               Integer32,
           rbridgeFibMacAddress
               RbridgeAddress
       }

   rbridgeFibNickname OBJECT-TYPE
       SYNTAX      RbridgeNickname
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
```

```
           "An RBridge nickname for which this RBridge has
           forwarding information."
       ::= { rbridgeUniFibEntry 1 }

   rbridgeFibPort OBJECT-TYPE
       SYNTAX      Integer32 (0..65535)
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "The port number of the port attached to the next-hop
           RBridge for the path towards the RBridge whose nickname
           is specified in this entry."
       ::= { rbridgeUniFibEntry 2 }

   rbridgeFibMacAddress OBJECT-TYPE
       SYNTAX      RbridgeAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The MAC address of the next-hop RBridge for the path
           towards the RBridge whose nickname is specified in this
           entry."
       ::= { rbridgeUniFibEntry 3 }

   rbridgeMultiFibTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF RbridgeMultiFibEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A table that contains information about egress nicknames
           used for multi-destination frame forwarding by this
           RBridge."
       ::= { rbridgeFdb 6 }

   rbridgeMultiFibEntry OBJECT-TYPE
       SYNTAX      RbridgeMultiFibEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A list of information about egress nicknames used for
           multi-destination frame forwarding by this RBridge."
       INDEX   { rbridgeMultiFibNickname }
       ::= { rbridgeMultiFibTable 1 }

   RbridgeMultiFibEntry ::=
       SEQUENCE {
           rbridgeMultiFibNickname
               RbridgeNickname,
```

```
            rbridgeMultiFibPorts
                PortList
        }

   rbridgeMultiFibNickname OBJECT-TYPE
       SYNTAX      RbridgeNickname
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "The nickname of the multicast distribution tree."
       ::= { rbridgeMultiFibEntry 1 }

   rbridgeMultiFibPorts OBJECT-TYPE
       SYNTAX      PortList
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The list of ports to which a frame destined to this
           multicast distribution tree is flooded. This may be pruned
           further based on other forwarding information."
       ::= { rbridgeMultiFibEntry 2 }


   -- ---------------------------------------------------------- --
   -- The RBridge VLAN Table
   -- ---------------------------------------------------------- --

   rbridgeVlanTable  OBJECT-TYPE
       SYNTAX      SEQUENCE OF RbridgeVlanEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A table that contains information about VLANs on the
           RBridge."
       ::= { rbridgeVlan 1 }

   rbridgeVlanEntry OBJECT-TYPE
       SYNTAX      RbridgeVlanEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A list of information about VLANs on the RBridge."
       INDEX   { dot1qVlanIndex }
       ::= { rbridgeVlanTable 1 }

   RbridgeVlanEntry ::=
       SEQUENCE {
           rbridgeVlanForwarderLost
```

```
                Counter32,
            rbridgeVlanDisableLearning
                TruthValue,
            rbridgeVlanSnooping
                INTEGER
        }

   rbridgeVlanForwarderLost OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The number of times this RBridge has lost appointed
           forwarder status for this VLAN on any of its ports."
       REFERENCE
           "RBridge section 4.8.2"
       ::= { rbridgeVlanEntry 1 }

   rbridgeVlanDisableLearning OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "Disable learning of MAC addresses seen in this VLAN.
           One application of this may be to restrict learning to
           ESADI. The default is false."
       REFERENCE
            "RBridge section 4.8"
       DEFVAL      { false }
       ::= { rbridgeVlanEntry 2 }

   rbridgeVlanSnooping OBJECT-TYPE
       SYNTAX      INTEGER {
                       notSupported(1),
                       ipv4(2),
                       ipv4v6(3)
                   }
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "IP Multicast Snooping on this VLAN. For RBridges
           performing both IPv4 and IPv6 IP Multicast Snooping, the
           value returned is ipv4v6(3)."
       REFERENCE
           "RBridge section 4.7"
       ::= { rbridgeVlanEntry 3 }

   -- ---------------------------------------------------------- --
```

```
   -- The RBridge VLAN Port Table
   -- ---------------------------------------------------------------- --

   rbridgeVlanPortTable  OBJECT-TYPE
       SYNTAX      SEQUENCE OF RbridgeVlanPortEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A table that contains information about VLANs on an RBridge
           port."
       ::= { rbridgeVlan 2 }

   rbridgeVlanPortEntry OBJECT-TYPE
       SYNTAX      RbridgeVlanPortEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A list of information about VLANs on the RBridge port."
       INDEX   { dot1qVlanIndex, rbridgeBasePort }
       ::= { rbridgeVlanPortTable 1 }

   RbridgeVlanPortEntry ::=
       SEQUENCE {
           rbridgeVlanPortInhibited
               TruthValue,
           rbridgeVlanPortForwarder
               TruthValue,
           rbridgeVlanPortAnnouncing
               TruthValue,
           rbridgeVlanPortDetectedVlanMapping
               TruthValue
       }

   rbridgeVlanPortInhibited OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This VLAN has been inhibited by the RBridge due to
           conflicting Forwarder information received from another
           RBridge."
       REFERENCE
           "RBridge section 4.2.4.3"
       ::= { rbridgeVlanPortEntry 1 }

   rbridgeVlanPortForwarder OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-only
```

```
       STATUS      current
       DESCRIPTION
           "This RBridge is an Appointed Forwarder for this VLAN on
           this port."
       REFERENCE
           "RBridge section 4.2.4.3"
       ::= { rbridgeVlanPortEntry 2 }

   rbridgeVlanPortAnnouncing OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "TRILL-Hellos tagged with this VLAN can be sent by this
           RBridge on this port. Defaults to true for enabled
           VLANs."
       REFERENCE
           "RBridge section 4.4.3"
       DEFVAL      { true }
       ::= { rbridgeVlanPortEntry 3 }

   rbridgeVlanPortDetectedVlanMapping OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "VLAN mapping has been detected on the link attached
           to this port."
       REFERENCE
           "RBridge section 4.4.5"
       ::= { rbridgeVlanPortEntry 4 }


   -- ---------------------------------------------------------------- --
   -- The RBridge Port Table
   -- ---------------------------------------------------------------- --

   rbridgePortCounterTable  OBJECT-TYPE
       SYNTAX      SEQUENCE OF RbridgePortCounterEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A table contains per-port counters for this RBridge."
       ::= { rbridgeCounter 1 }

   rbridgePortCounterEntry OBJECT-TYPE
       SYNTAX      RbridgePortCounterEntry
       MAX-ACCESS  not-accessible
```

```
        STATUS      current
        DESCRIPTION
            "Counters for a port on this RBridge."
        INDEX   { rbridgeBasePort }
        ::= { rbridgePortCounterTable 1 }

    RbridgePortCounterEntry ::=
        SEQUENCE {
            rbridgePortRpfChecksFailed
                Counter32,
            rbridgePortHopCountsExceeded
                Counter32,
            rbridgePortOptions
                Counter32,
            rbridgePortTrillInFrames
                Counter32,
            rbridgePortTrillOutFrames
                Counter32,
            rbridgePortTrillInOverflowFrames
                Counter32,
            rbridgePortTrillOutOverflowFrames
                Counter32
        }

    rbridgePortRpfChecksFailed OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The number of times a multidestination frame was
            dropped on this port because the RPF check failed."
        REFERENCE
            "RBridge section 4.5.2"
        ::= { rbridgePortCounterEntry 1 }

    rbridgePortHopCountsExceeded OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The number of times a frame was dropped on this port
            because its hop count was zero."
        REFERENCE
            "RBridge section 3.6"
        ::= { rbridgePortCounterEntry 2 }

    rbridgePortOptions OBJECT-TYPE
        SYNTAX      Counter32
```

```
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The number of times a frame was dropped on this port
           because it contained unsupported options."
       REFERENCE
           "RBridge section 3.5"
       ::= { rbridgePortCounterEntry 3 }

   rbridgePortTrillInFrames OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The number of TRILL-encapsulated frames that have been
           received by this port from its attached link, including
           management frames."
       REFERENCE
           "RBridge section 2.3"
       ::= { rbridgePortCounterEntry 4 }

   rbridgePortTrillOutFrames OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The number of TRILL-encapsulated frames that have been
           transmitted by this port to its attached link, including
           management frames."
       REFERENCE
           "RBridge section 2.3"
       ::= { rbridgePortCounterEntry 5 }

   rbridgePortTrillInOverflowFrames OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The number of times the rbridgePortTrillInFrames
           counter on this port has overflowed."
       ::= { rbridgePortCounterEntry 6 }

   rbridgePortTrillOutOverflowFrames OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The number of times the rbridgePortTrillOutFrames
```

```
            counter on this port has overflowed."
        ::= { rbridgePortCounterEntry 7 }


   -- ---------------------------------------------------------- --
   -- The RBridge VLAN ESADI Table
   -- ---------------------------------------------------------- --

   rbridgeEsadiTable  OBJECT-TYPE
       SYNTAX      SEQUENCE OF RbridgeEsadiEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A table that contains information about ESADI instances on
           VLANs, if available."
       REFERENCE
           "RBridge section 4.2.5"
       ::= { rbridgeEsadi 1 }

   rbridgeEsadiEntry OBJECT-TYPE
       SYNTAX      RbridgeEsadiEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Information about an ESADI instance on a VLAN."
       INDEX   { dot1qVlanIndex }
       ::= { rbridgeEsadiTable 1 }

   RbridgeEsadiEntry ::=
       SEQUENCE {
           rbridgeEsadiStatus
               INTEGER,
           rbridgeEsadiConfidence
               Integer32,
           rbridgeEsadiDrbPriority
               Integer32,
           rbridgeEsadiDrb
               RbridgeAddress,
           rbridgeEsadiDrbHoldingTime
               Integer32
       }

   rbridgeEsadiStatus OBJECT-TYPE
       SYNTAX      INTEGER {
                       enabled(1),
                       disabled(2),
                       delete(3)
                   }
```

```
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "If the RBridge is participating in an ESADI instance for
           this VLAN, the default value is enabled(1). To delete this
           instance, the value delete(3) is written to this object."
       REFERENCE
           "RBridge section 4.2.5"
       DEFVAL      { enabled }
       ::= { rbridgeEsadiEntry 1 }

   rbridgeEsadiConfidence OBJECT-TYPE
       SYNTAX      Integer32 (0..254)
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "Confidence level of address entries sent by this
           ESADI. The default is 16."
       REFERENCE
           "RBridge section 4.2.5"
       DEFVAL      { 16 }
       ::= { rbridgeEsadiEntry 2 }

   rbridgeEsadiDrbPriority OBJECT-TYPE
       SYNTAX      Integer32 (0..127)
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "The priority of this RBridge for being selected as
           DRB for this ESADI instance."
       REFERENCE
           "RBridge section 4.2.5"
       ::= { rbridgeEsadiEntry 3 }

   rbridgeEsadiDrb OBJECT-TYPE
       SYNTAX      RbridgeAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The DRB on this ESADI instance's virtual link."
       REFERENCE
           "RBridge section 4.2.5"
       ::= { rbridgeEsadiEntry 4 }

   rbridgeEsadiDrbHoldingTime OBJECT-TYPE
       SYNTAX      Integer32(0..127)
       MAX-ACCESS  read-create
       STATUS      current
```

```
        DESCRIPTION
            "The holding time for this ESADI instance."
        REFERENCE
            "RBridge section 4.2.5"
        ::= { rbridgeEsadiEntry 5 }


    -- ---------------------------------------------------------------- --
    -- The RBridge IP Multicast Snooping Port Table
    -- ---------------------------------------------------------------- --

    rbridgeSnoopingPortTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF RbridgeSnoopingPortEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "For Rbridges implementing IP Multicast Snooping,
             information about ports on which the presence of IPv4
            or IPv6 Multicast Routers has been detected."
        REFERENCE
            "RBridge section 4.7"
        ::= { rbridgeSnooping 1 }

    rbridgeSnoopingPortEntry OBJECT-TYPE
        SYNTAX      RbridgeSnoopingPortEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Information about ports on which the presence of IPv4
            or IPv6 Multicast Routers has been detected."
        INDEX   { rbridgeBasePort }
        ::= { rbridgeSnoopingPortTable 1 }

    RbridgeSnoopingPortEntry ::=
        SEQUENCE {
            rbridgeSnoopingPortAddrType
                InetAddressType,
            rbridgeSnoopingPortAddr
                InetAddress
        }

    rbridgeSnoopingPortAddrType OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The IP address type of an IP multicast router detected
            on this port."
```

```
       REFERENCE
           "RBridge section 4.7"
       ::= { rbridgeSnoopingPortEntry 1 }

   rbridgeSnoopingPortAddr OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The IP address of an IP multicast router detected on
           this port."
       REFERENCE
           "RBridge section 4.7"
       ::= { rbridgeSnoopingPortEntry 2 }

   -- ---------------------------------------------------------- --
   -- The RBridge IP Multicast Snooping Address Table
   -- ---------------------------------------------------------- --

   rbridgeSnoopingAddrTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF RbridgeSnoopingAddrEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "For Rbridges implementing IP Multicast Snooping,
            information about IP Multicast addresses being
            snooped."
       REFERENCE
           "RBridge section 4.8"
       ::= { rbridgeSnooping 2 }

   rbridgeSnoopingAddrEntry OBJECT-TYPE
       SYNTAX      RbridgeSnoopingAddrEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Information about IP Multicast addresses being
            snooped."
       INDEX  { dot1qVlanIndex, rbridgeSnoopingAddrType,
                rbridgeSnoopingAddr }
       ::= { rbridgeSnoopingAddrTable 1 }

   RbridgeSnoopingAddrEntry ::=
       SEQUENCE {
           rbridgeSnoopingAddrType
               InetAddressType,
           rbridgeSnoopingAddr
               InetAddress,
```

```
            rbridgeSnoopingAddrPorts
                PortList
        }

    rbridgeSnoopingAddrType OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The IP multicast address type for which a listener has been
            detected by this RBridge."
        REFERENCE
            "RBridge section 4.7"
        ::= { rbridgeSnoopingAddrEntry 1 }

    rbridgeSnoopingAddr OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The IP multicast address for which a listener has been
            detected by this RBridge."
        REFERENCE
            "RBridge section 4.7"
        ::= { rbridgeSnoopingAddrEntry 2 }

    rbridgeSnoopingAddrPorts OBJECT-TYPE
        SYNTAX      PortList
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The set of ports on which a listener has been detected
            for this IP multicast address."
        REFERENCE
            "RBridge section 4.7"
        ::= { rbridgeSnoopingAddrEntry 3 }


    -- ---------------------------------------------------------- --
    -- Distribution Trees
    -- ---------------------------------------------------------- --

    rbridgeDtreePriority OBJECT-TYPE
        SYNTAX      Integer32 (1..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The Distribution tree root priority for this Rbridge.
```

```
             The default value of this object is 32768."
        REFERENCE
            "RBridge section 4.5"
        ::= { rbridgeDtree 1 }

    rbridgeDtreeActiveTrees OBJECT-TYPE
        SYNTAX      Integer32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The total number of trees being computed by all Rbridges
            campus."
        REFERENCE
            "RBridge section 4.5"
        ::= { rbridgeDtree 2 }

    rbridgeDtreeMaxTrees OBJECT-TYPE
        SYNTAX      Integer32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The maximum number of trees this Rbridge can compute."
        REFERENCE
            "RBridge section 4.5"
        ::= { rbridgeDtree 3 }

    rbridgeDtreeDesiredUseTrees OBJECT-TYPE
        SYNTAX      Integer32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The maximum number of trees this Rbridge would like to
            use for transmission of ingress multi-destination frames."
        REFERENCE
            "RBridge section 4.5"
        ::= { rbridgeDtree 4 }

    rbridgeDtreeTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF RbridgeDtreeEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Information about Distribution Trees being computed
            by this Rbridge."
        REFERENCE
            "RBridge section 4.5"
        ::= { rbridgeDtree 5 }
```

```
   rbridgeDtreeEntry OBJECT-TYPE
       SYNTAX      RbridgeDtreeEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "List of information about Distribution Trees being computed
           by this Rbridge."
       INDEX  { rbridgeDtreeNumber }
       ::= { rbridgeDtreeTable 1 }

   RbridgeDtreeEntry ::=
       SEQUENCE {
           rbridgeDtreeNumber
               Integer32,
           rbridgeDtreeNick
               RbridgeNickname,
           rbridgeDtreeIngress
               TruthValue
       }

   rbridgeDtreeNumber OBJECT-TYPE
       SYNTAX      Integer32 (0..65535)
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "The tree number of a distribution tree being computed by
           this RBridge."
       REFERENCE
           "RBridge section 4.5"
       ::= { rbridgeDtreeEntry 1 }

   rbridgeDtreeNick OBJECT-TYPE
       SYNTAX      RbridgeNickname
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The nickname of the distribution tree."
       REFERENCE
           "RBridge section 4.5"
       ::= { rbridgeDtreeEntry 2 }

   rbridgeDtreeIngress OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "Indicates whether this RBridge might choose this
           distribution tree to ingress a multi-destination frame."
```

```
        REFERENCE
            "RBridge section 4.5"
        ::= { rbridgeDtreeEntry 3 }


    -- ---------------------------------------------------------------- --
    -- TRILL neighbor list
    -- ---------------------------------------------------------------- --

    rbridgeTrillMinMtuDesired OBJECT-TYPE
        SYNTAX      Integer32
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The desired minimum acceptable inter-RBridge link MTU for
            the campus, that is, originatingLSPBufferSize. The default
            is 1470 bytes."
        REFERENCE
            "RBridge section 4.3"
        ::= { rbridgeTrill 1 }

    rbridgeTrillSz OBJECT-TYPE
        SYNTAX      Integer32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The minimum acceptable inter-Rbridge link size for the
            campus for the proper operation of TRILL IS-IS."
        REFERENCE
            "RBridge section 4.3"
        ::= { rbridgeTrill 2 }

    rbridgeTrillMaxMtuProbes OBJECT-TYPE
        SYNTAX      Integer32 (1..255)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The number of failed MTU-probes before the RBridge
            concludes that a particular MTU is not supported by
            a neighbor. The default is 3."
        REFERENCE
            "RBridge section 4.3"
        ::= { rbridgeTrill 3 }

    rbridgeTrillNbrTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF RbridgeTrillNbrEntry
        MAX-ACCESS  not-accessible
        STATUS      current
```

```
        DESCRIPTION
            "Information about this Rbridge's TRILL neighbors."
        REFERENCE
            "RBridge section 4.4.2.1"
        ::= { rbridgeTrill 4 }

    rbridgeTrillNbrEntry OBJECT-TYPE
        SYNTAX      RbridgeTrillNbrEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "List of information about this Rbridge's TRILL neighbors."
        INDEX  { rbridgeTrillNbrMacAddr }
        ::= { rbridgeTrillNbrTable 1 }

    RbridgeTrillNbrEntry ::=
        SEQUENCE {
            rbridgeTrillNbrMacAddr
                MacAddress,
            rbridgeTrillNbrMtu
                Integer32,
            rbridgeTrillNbrFailedMtuTest
                TruthValue
        }

    rbridgeTrillNbrMacAddr OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The MAC address of a neighbor of this RBridge."
        REFERENCE
            "RBridge section 4.4.2.1"
        ::= { rbridgeTrillNbrEntry 1 }

    rbridgeTrillNbrMtu OBJECT-TYPE
        SYNTAX      Integer32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "MTU size to this neighbor for IS-IS communication purposes."
        REFERENCE
            "RBridge section 4.3.2"
        ::= { rbridgeTrillNbrEntry 2 }

    rbridgeTrillNbrFailedMtuTest OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
```

```
        STATUS      current
        DESCRIPTION
            "If true, indicates that the neighbor's tested MTU is less
            than the minimum acceptable inter-bridge link MTU for the
            campus (1470)."
        REFERENCE
            "RBridge section 4.3.1"
        ::= { rbridgeTrillNbrEntry 3 }


    -- ---------------------------------------------------------- --
    -- Notifications for use by RBridges
    -- ---------------------------------------------------------- --

    rbridgeBaseNewDrb NOTIFICATION-TYPE
        -- OBJECTS     { }
        STATUS      current
        DESCRIPTION
            "The RBridgeBaseNewDrb trap indicates that the sending agent
            has become the new Designated RBridge; the trap is
            sent by an RBridge soon after its election as the new DRB
            root, e.g., upon expiration of the Topology Change Timer,
            immediately subsequent to its election.  Implementation
            of this trap is optional."
        ::= { rbridgeNotifications 1 }

    rbridgeBaseTopologyChange NOTIFICATION-TYPE
        -- OBJECTS     { }
        STATUS      current
        DESCRIPTION
            "RBridgeBaseTopologyChange trap is sent by an RBridge when
            any of its configured ports transitions to/from Vlan-x
            designated forwarder.  The trap is not sent if a newDrb
            trap is sent for the same transition.  Implementation of
            this trap is optional."
        ::= { rbridgeNotifications 2 }

-- Compliance and Group sections

    rbridgeGroup          OBJECT IDENTIFIER ::= { rbridgeConformance 1 }

    rbridgeCompliances    OBJECT IDENTIFIER ::= { rbridgeConformance 2 }


    -- ---------------------------------------------------------- --
    -- Units of Conformance
    -- ---------------------------------------------------------- --
```

```
   rbridgeBaseGroup OBJECT-GROUP
       OBJECTS {
           rbridgeBaseTrillVersion,
           rbridgeBaseNumPorts,
           rbridgeBaseForwardDelay,
           rbridgeBaseUniMultipathEnable,
           rbridgeBaseMultiMultipathEnable,
           rbridgeBaseNicknameNumber,
           rbridgeBaseAcceptEncapNonadj
       }
       STATUS      current
       DESCRIPTION
           "A collection of objects providing basic control
           and status information for the RBridge."
       ::= { rbridgeGroup 1 }

   rbridgeBaseNicknameGroup OBJECT-GROUP
       OBJECTS {
           rbridgeBaseNicknamePriority,
           rbridgeBaseNicknameDtrPriority,
           rbridgeBaseNicknameStatus
       }
       STATUS      current
       DESCRIPTION
           "A collection of objects providing basic control
           and status information for RBridge nicknames."
       ::= { rbridgeGroup 2 }

   rbridgeBasePortGroup OBJECT-GROUP
       OBJECTS {
           rbridgeBasePortIfIndex,
           rbridgeBasePortDisable,
           rbridgeBasePortTrunkPort,
           rbridgeBasePortAccessPort,
           rbridgeBasePortP2pHellos,
           rbridgeBasePortState,
           rbridgeBasePortDesiredDesigVlan,
           rbridgeBasePortDesigVlan,
           rbridgeBasePortInhibitionTime,
           rbridgeBasePortDisableLearning,
           rbridgeBasePortStpRoot,
           rbridgeBasePortStpRootChanges,
           rbridgeBasePortStpWiringCloset
       }
       STATUS      current
       DESCRIPTION
           "A collection of objects providing basic control
           and status information for RBridge ports."
```

```
        ::= { rbridgeGroup 3 }

   rbridgeFdbGroup OBJECT-GROUP
       OBJECTS {
           rbridgeConfidenceNative,
           rbridgeConfidenceDecap,
           rbridgeConfidenceStatic,
           rbridgeUniFdbPort,
           rbridgeUniFdbNick,
           rbridgeUniFdbConfidence,
           rbridgeUniFdbStatus
       }
       STATUS      current
       DESCRIPTION
           "A collection of objects providing information
           about the Unicast Address Database."
       ::= { rbridgeGroup 4 }

   rbridgeFibGroup OBJECT-GROUP
       OBJECTS {
           rbridgeFibMacAddress,
           rbridgeMultiFibPorts
       }
       STATUS      current
       DESCRIPTION
           "A collection of objects providing information
           about the Unicast and Multicast FIBs."
       ::= { rbridgeGroup 5 }

   rbridgeVlanGroup OBJECT-GROUP
       OBJECTS {
           rbridgeVlanForwarderLost,
           rbridgeVlanDisableLearning,
           rbridgeVlanSnooping,
           rbridgeVlanPortInhibited,
           rbridgeVlanPortForwarder,
           rbridgeVlanPortAnnouncing,
           rbridgeVlanPortDetectedVlanMapping
       }
       STATUS      current
       DESCRIPTION
           "A collection of objects providing information
           about VLANs on the RBridge."
       ::= { rbridgeGroup 6 }

   rbridgePortCoounterGroup OBJECT-GROUP
       OBJECTS {
```

```
            rbridgePortRpfChecksFailed,
            rbridgePortHopCountsExceeded,
            rbridgePortOptions,
            rbridgePortTrillInFrames,
            rbridgePortTrillOutFrames,
            rbridgePortTrillInOverflowFrames,
            rbridgePortTrillOutOverflowFrames
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing per-port
            counters for the RBridge."
        ::= { rbridgeGroup 7 }

    rbridgeEsadiGroup OBJECT-GROUP
        OBJECTS {
            rbridgeEsadiStatus,
            rbridgeEsadiConfidence,
            rbridgeEsadiDrbPriority,
            rbridgeEsadiDrb,
            rbridgeEsadiDrbHoldingTime
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information
            about ESADI instances on the RBridge."
        ::= { rbridgeGroup 8 }

    rbridgeSnoopingGroup OBJECT-GROUP
        OBJECTS {
            rbridgeSnoopingPortAddrType,
            rbridgeSnoopingPortAddr,
            rbridgeSnoopingAddrPorts
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information
            about IP Multicast Snooping."
        ::= { rbridgeGroup 9 }

    rbridgeDtreeGroup OBJECT-GROUP
        OBJECTS {
            rbridgeDtreePriority,
            rbridgeDtreeActiveTrees,
            rbridgeDtreeMaxTrees,
            rbridgeDtreeDesiredUseTrees,
            rbridgeDtreeNick,
            rbridgeDtreeIngress
```

```
       }
       STATUS      current
       DESCRIPTION
           "A collection of objects providing information
           about Distribution Trees."
       ::= { rbridgeGroup 10 }

   rbridgeTrillGroup OBJECT-GROUP
       OBJECTS {
           rbridgeTrillMinMtuDesired,
           rbridgeTrillSz,
           rbridgeTrillMaxMtuProbes,
           rbridgeTrillNbrMtu,
           rbridgeTrillNbrFailedMtuTest
       }
       STATUS      current
       DESCRIPTION
           "A collection of objects providing information
           about TRILL neighbors."
       ::= { rbridgeGroup 11 }

   rbridgeNotificationGroup NOTIFICATION-GROUP
       NOTIFICATIONS {
           rbridgeBaseNewDrb,
           rbridgeBaseTopologyChange
       }
       STATUS      current
       DESCRIPTION
           "A collection of objects describing notifications (traps)."
       ::= { rbridgeGroup 12 }



   -- ---------------------------------------------------------- --
   -- Compliance Statement
   -- ---------------------------------------------------------- --

   rbridgeCompliance MODULE-COMPLIANCE
          STATUS      current
          DESCRIPTION
              "The compliance statement for support of RBridge
              services."

          MODULE
              MANDATORY-GROUPS {
                  rbridgeBaseGroup,
                  rbridgeBaseNicknameGroup,
                  rbridgeBasePortGroup,
```

```
                  rbridgeFdbGroup,
                  rbridgeFibGroup,
                  rbridgeVlanGroup,
                  rbridgeDtreeGroup,
                  rbridgeTrillGroup
              }

          GROUP   rbridgePortCoounterGroup
          DESCRIPTION
              "Implementation of this group is optional."

          GROUP   rbridgeEsadiGroup
          DESCRIPTION
              "Implementation of this group is optional."

          GROUP   rbridgeSnoopingGroup
          DESCRIPTION
              "Implementation of this group is optional."

          GROUP   rbridgeNotificationGroup
          DESCRIPTION
              "Implementation of this group is optional."

          ::= { rbridgeCompliances 1 }


      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 a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or 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 a negative effect on network operations. These tables and objects and their sensitivity/vulnerability are described below.

The following tables and objects in the RBRIDGE-MIB can be manipulated to interfere with the operation of RBridges:

- rbridgeBaseUniMultipathEnable affects the ability of the RBridge to multipath unicast traffic, and rbridgeBaseMultiMultipathEnable affects the ability of the Rbridge to multipath multi-destination traffic.

- rbridgeBasePortTable contains a number of objects that may affect network connectivity. Actions that may be triggered by manipulating objects in this table include disabling of an RBridge port; discarding of native packets; disabling learning and others.

- rbridgeEsadiTable contains objects that affect the operation of the ESADI protocol used for learning, and manipulation of the objects contained therein can be used to confuse the learning ability of Rbridges.

- rbridgeDtreePriority can affect computation of distribution trees within an Rbridge campus, thereby affecting forwarding of multi-destination traffic.

- rbridgeTrillMinMtuDesired can affect the size of packets being used to exchange information between RBridges.

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 even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. For example, access to network topology and Rbridge attributes can reveal information that should not be available to all users of the network.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), 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 mechanisms (for authentication and privacy).

Further, deployment of SNMP versions 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.

For other RBridge security considerations see [RBridge].

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

---------- ----------------------

rbridgeMIB { 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. Contributors

The authors would like to acknowledge the contributions of Donald Eastlake, Radia Perlman and Anoop Ghanwani. We invite you to join the mailing list at http://www.postel.org/rbridge.

11. References

11.1 Normative References

[RBridge] Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "RBridges: Base Protocol Specification", Work in Progress, January 2010.

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

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J.

Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC4188] Norseth, K. and E. Bell, "Definitions of Managed Objects for Bridges", RFC 4188, September 2005.

[RFC4363] Levi, D. and D. Harrington, "Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering, and Virtual LAN Extensions", RFC 4363, January 2006.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC4444] Parker, J., "Management Information Base for Intermediate System to Intermediate System (IS-IS)", RFC 4444, April 2006.

[802.1Q-2005] Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks", IEEE 802.1Q, May 2006.

11.2 Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC5556] Touch, J. and R. Perlman, "Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement", RFC 5556, May 2009.

Appendix A. Change Log

Note to RFC Editor: Please remove this appendix before publication as an RFC.

Changes from -01 to -02
1. Added rbridgeTrillSz, campus-wide minimum MTU
2. Added DEFAULT clause to read-create objects
3. Added references to IEEE 802.1 MIBs
4. Changed base MIB structure to group MIB objects under one sub-tree
5. Fixed errors and warnings reported by libsmi compiler

6. Enhanced detail in Security Considerations section.

Authors' Addresses

Anil Rijhsinghani
Hewlett-Packard Networking
350 Campus Drive
Marlboro, MA
USA

Phone: +1 508 323 1251
EMail: anil@charter.net

Kate Zebrose
H.W. Embedded
26 Josephine Ave
Somerville, MA
USA

Phone: +1 617 840 9673
EMail: kate.zebrose@alum.mit.edu

Copyright and License Notice

TRILL Working Group Donald Eastlake
INTERNET-DRAFT Huawei
Intended status: Proposed Standard Anoop Ghanwani
Brocade
Vishwas Manral
IP Infusion
Caitlin Bestler
Quantum
Expires: September 9, 2011 March 10, 2011

RBridges: TRILL Header Options
<draft-ietf-trill-rbridge-options-04.txt>

Abstract

The TRILL base protocol standard specifies minimal hooks to safely support TRILL Header options. This draft specifies the format for options and some initial options.

Status of This Memo

Table of Contents

1. Introduction

The base TRILL protocol standard [RFCtrill] provides a TRILL Header options feature and describes minimal hooks to safely support that feature. But, except for the first two bits, it does not specify the structure of the options extension to the TRILL Header nor the details of any particular options. This draft specifies that format and some initial options: a special Flow ID field, ECN (Explicit Congestion Notification) extended header flags, and a test/pad option.

Section 2 below describes the general principles of operation, format, and ordering of TRILL Header Options. Other than the special Flow ID option, TRILL Header options are of two kinds: extended header flags and TLV (Type, Length, Value) encoded options.

Section 3 describes a specific extended flag option while Section 4 describes a specific TLV encoded option.

1.1 Conventions used in this document

The terminology and acronyms defined in [RFCtrill] are used herein with the same meaning.

In this documents, "IP" refers to both IPv4 and IPv6.

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

2. TRILL Header Options

The base TRILL Protocol includes an option feature for extension of the TRILL Header (see [RFCtrill] Sections 3.5 and 3.8). The 5-bit Op-Length header field gives the length of the extension to the TRILL Header in units of 4 octets, which allows up to 124 octets of header extension. If Op-Length is zero there is no header extension present; else, this area follows immediately after the Ingress Rbridge Nickname field of the TRILL Header. The optional extensions area consists of an extended flags area possibly followed by TLV options. Each TLV option present is 32-bit aligned. There is a special Flow ID option that may also occur in the extended flags area.

As described below, provision is made for both hop-by-hop options, which might affect any RBridge that receives a TRILL Data frame containing such an extension, and ingress-to-egress options, which would only necessarily affect the RBridge(s) where a TRILL frame is decapsulated. Provision is also made for both "critical" and "non-critical" options. Any RBridge receiving a frame with a critical hop-by-hop option that it does not implement MUST discard the frame because it is unsafe to process the frame without understanding the critical option. Any egress RBridge receiving a frame with a critical ingress-to-egress option it does not implement MUST drop the frame if it is a known unicast frame; if it is a multi-destination TRILL Data frame with a critical ingress-to-egress option that the RBridge does not implement, then it MUST NOT be egressed at that RBridge but it is still forwarded on the distribution tree. Non-critical options can be safely ignored.

Any option indicating a significant change in the structure or interpretation of later parts of the frame which, if the option were ignored, could reasonably cause a failure of service or violation of security policy MUST be a critical option. If such an extension affects any fields that transit RBridges will examine, it MUST be a hop-by-hop critical option.

TLV options also have a "mutability" flag that has a different meaning for ingress-to-egress and for hop-by-hop.

For an ingress-to-egress option, the mutability flag indicates whether the value associated with the option can change at a transit RBridge (mutable options) or cannot so change (immutable options). For example, an ingress-to-egress security option could protect the value of an immutable ingress-to-egress option. But such a security option generally could not protect a mutable value as a transit RBridge could change that value but might not have the keys to recompute a signature or authentication code to take a changed value into account.

For a non-critical hop-by-hop option, the mutability flag indicates whether a transit RBridge that does not implement the option is permitted (mutable) or not permitted (immutable) to remove the option. A transit RBridge is not required to remove a hop-by-hop option that it does not implement.

For critical hop-by-hop options, the mutability flag is meaningless. If the RBridge does not implement the critical hop-by-hop option, it MUST drop the frame. If it does implement the critical hop-by-hop option, it will know whether or not it may/should/must remove it. For critical hop-by-hop options, the mutability flag is set to zero ("immutable") on transmission and ignored on receipt.

> Note: Most RBridges implementations are expected to be optimized for simple and common cases of frame forwarding and processing. Although the hard limit on the header options area length, the 32-bit alignment of TLV options, and the presence of critical option summary bits, as described below, are intended to assist in the efficient hardware based processing of frames with a TRILL header options area, nevertheless the inclusion of options, particularly TLV options, may cause frame processing using a "slow path" with inferior performance to "fast path" processing. Limited slow path throughput of such frames could cause them to be discarded.

2.1 RBridge Option Handling Requirements

The requirements given in this section are in addition to the option handling requirements in [RFCtrill].

All RBridges MUST be able to check whether there are any critical options present that are necessarily applicable to their processing of the frame as detailed below. If they do not implement all such critical options present, they MUST discard the frame or, in some circumstances as described above for certain multi-destination frames, continue to forward the frame but MUST NOT egress the frame.

Transit RBridges MUST transparently forward all immutable ingress-to-egress header options in frames that they forward. Any changes made by a transit RBridge to a mutable ingress-to-egress option value MUST be a change permitted by the specification of that option.

In addition, a transit RBridge:

- o MAY add, if space is available, or remove hop-by-hop options as specified for such options;
- o MAY change the value and/or length of a mutable ingress-to-egress TLV option as permitted by that option’s specification and provided there is enough room if lengthening it;

- o MUST adjust the length of the options area, including changing Op-Length in the TRILL header, as appropriate for any changes it has made;
- o MUST NOT add, remove, or re-order ingress-to-egress options.
- o with regard to any non-critical hop-by-hop options that the transit RBridge does not implement, it MAY remove them if they are mutable but MUST transparently copy them when forwarding a frame if they are immutable.

2.2 No Critical Surprises

RBridges advertise the ingress-to-egress options they support in their IS-IS LSP and advertise the hop-by-hop options they support at a port on the link connected to that port. An RBridge is not required to support any options.

Unless an RBridge advertises support for a critical option, it will not normally receive frames with that option.

An RBridge SHOULD NOT add a critical option to a frame unless,
- for a critical hop-by-hop option, it has determined that the next hop RBridge or RBridges that will accept the frame support that option, or
- for a critical ingress-to-egress option, it has determined that the RBridge or RBridges that will egress the frame support that option.

"SHOULD NOT" is specified since there may be cases where it is acceptable for those frames, particularly for the multi-destination case, to be discarded by any RBridges that do not implement the option.

2.3 Options Format

If any options are present in a TRILL Header, as indicated by a non-zero Op-Length field, the first 32 or 64 bits of the options area consist of extended header flags and the Flow ID, as described below. The remainder of the options area, if any, after this initial 32 or 64 bits, consists of TLV (Type Length Value) options aligned on 32-bit boundaries. Section 2.3.2 specifies the format of a TLV option. Section 2.3.3 describes the marshaling of TLV options.

2.3.1 Extended Header Flags Area

The first 32 bits of the Options Area are organized as follows:

```
| 0    1   2   3-4  5-7  8-10 11-12  13   14   15 | 16 - 31 |
+----+----+----+----+----+----+-----+----+----+----+---------+
|CHbH|CItE|MEF |CHHF|NHHF|CIEF|NIEF |NHHT|CIET|NIET| Flow ID |
+----+----+----+----+----+----+-----+----+----+----+---------+
```

Figure 1: Options Area Initial 32 Bits

Any RBridge adding an options area to a TRILL Header must set these 32 bits to zero except when permitted or required to set one or more of them as specified. The meanings of these bits are listed in the table below and then further described.

Bit(s)	Description
0	CHbH: Critical Hop-by-Hop option(s) are present.
1	CItE: Critical Ingress-to-Egress option(s) are present.
2	MEF: More Extended Flags, indicates that an additional 32-bit extended flags area is present as described below.
3-4	CHHF: Critcial Hob-by-Hop extended Flag bits.
5-7	NHHF: Non-critical Hop-by-Hop extended Flag bits.
8-10	CIEF: Critical Ingress-to-Egress extended Flag bits.
11-12	NIEF: Non-critical Ingress-to-Egress extended Flag bits.
13	NHHT: Non-critical Hop-by-Hop TLV option(s) are present.
14	CIET: Critical Ingress-to-Egress TLV option(s) are present.
15	NIET: Non-critical Ingress-to-Egress TLV option(s) are present.
16-31	Flow ID if non-zero.

All extended flags are considered mutable except the critical hop-by-hop extended flags.

For TRILL Data frames with options present, any transit RBridge MUST transparently copy bits 8 through 12, except as permitted by an option implemented by that RBridge, but MAY either copy or clear any of the bits 5 through 7. Even if a transit RBridge removes all TLV options from a TRILL Header when allowed to do so, it MUST NOT eliminate the options area in a forwarded frame if any of bits 3, 4, or 8 through 12 remain non-zero; however, if there are no TLV options and all of bits 2 through 31 are zero, then the summary bits will also be zero and the transit RBridge MAY eliminate the Options area in the frame, setting Op-Length to zero.

2.3.1.1 Critical Summary Bits

The top two bits of the options area, bits 0 and 1 above, are called the critical summary bits. They summarize the presence of critical options as follows:

CHbH: If the CHbH (Critical Hop by Hop) bit is one, one or more critical hop-by-hop options are present in the options area. Transit RBridges that do not support all of the critical hop-by-hop options present, for example an RBridge that supported no hop-by-hop options, MUST drop the frame. If the CHbH bit is zero, the frame is safe, from the point of view of options processing, for a transit RBridge to forward, regardless of what options that RBridge does or does not support. A transit RBridge that supports none of the options present MUST transparently forward the options area when it forwards a frame, except that it MAY remove mutable hop-by-hop options.

CItE: If the CItE (Critical Ingress to Egress) bit is a one, one or more critical ingress-to-egress options are present in the options area. If it is zero, no such options are present. If either CHbH or CItE is non-zero, egress RBridges that do not support all critical options present, for example an RBridge that supports no options, MUST drop the frame. If both CHbH and CItE are zero, the frame is safe, from the point of view of options, for any egress RBridge to process, regardless of what options that RBridge does or does not support.

The critical summary bits enable efficient processing of TRILL Data frames by RBridges that support no critical options and by transit RBridges that support no critical hop-by-hop options. Such RBridges need only check whether Op-Length is non-zero and, if it is, the top one or two bits just after the fixed portion of the TRILL Header.

2.3.1.2 MEF, More Extended Flags

Bit 2, if set, indicates there are an additional 32 bits of extended flags. They are organized as shown below. The start of the TLV options, if any, is moved to after these additional bit options.

```
 |    32 - 39    |    40 - 47    |    48 - 55    |    56 - 63    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Critical HbH  |NonCritical HbH| Critical ItE  |NonCritical ItE|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Extended Flag Bits 32 to 63

2.3.1.3 Specific Initial Bit Extended Flags

CHHB, bits 3 and 4, are Critical Hob-by-Hop Bits.

NHHB, bits 5 through 7, are Non-critical Hop-by-Hop Bits.

CIEB, bits 8 through 10, are Critical Ingress-to-Egress Bits.

NIEB, bits 11 and 12, are Non-critical Ingress-to-Egress Bits.

The bits above are available for indicating extended header flags, except for two NHHF allocated by Section 3.1 below.

2.3.1.4 TLV Summary Bits

It is anticipated that in most cases the interpretation of TLV encoded options in TRILL data frames will be handled by slow path software. To minimize unnecessary resort to the slow path, the TLV summary bits, plus a special check for critical hop-by-hop TLV options, enable an RBridge to quickly determine if any TLV encoded options of the category or categories it implements are present.

Bits 13-15, the NHHT, CIET, and NIET bits, indicate the presence later in the TRILL Header of TLV encoded Non-critical Hop-by-Hop, Critical Ingress-to-Egress, and Non-critical Ingress-to-Egress TLV options respectively.

There is no Critical Hop-by-Hop TLV flag bit because the presence of one or more such TLV options can be determined by examining Op-Length and, if Op-Length and the MEF bit indicate that there are TLV options beyond the extended flags area, examining the top two bits of the first options area byte after the extended flags area. The ordering restrictions on TLV options require that, if any Critical Hop-by-Hop TLV options are present, the appear first in the TLV options area. Thus it is adequate to check only if the first TLV option present is a Critical Hop-by-Hop option, which can be determined from the top two bits of its first byte.

2.3.1.5 Flow ID

In connection with the multi-pathing of frames, frames that are part of the same order-dependent flow need to follow the same path. Methods to determine flows are beyond the scope of the this document; however, it may be useful, once the flow of a unicast frame has been determined, to preserve and transmit that information for use by subsequent RBridges.

The Flow ID option is a specially encoded non-critical hop-by-hop option that appears in bits 16 through 31 of the initial bit encoded options area. Its presence is indicated by a non-zero value in that field.

It is considered hop-by-hop because it can be added or changed by a transit RBridge and transit RBridges can use it to make forwarding decisions. Because the ingress RBridge may know the most about a frame, it is expected that this option would most commonly be added at the ingress RBridge. Once set non-zero in a frame, the option SHOULD NOT be removed, set to zero, or changed unless, for example, a campus is divided into regions such that different Flow IDs would make sense in different regions.

2.3.2 TLV Option Format

TRILL Header options, other than the extended header flags and Flow ID described above, are TLV encoded, with some flag bits in the Type and Length octets, in the format show in Figure 3.

```
| 0  1  2  3  4  5  6  7  8  9|10|    11-15     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+---
|IE|NC|      Type             |MT|   Length     | value...
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+---
```

Figure 3. Option TLV Structure

The highest order bit of the first octet (IE) is zero for hop-by-hop options and one for ingress-to-egress options. Hop-by-hop options are potentially applicable to every RBridge that receives the frame. Ingress-to-egress options are only inserted at the ingress RBridge and are applicable at egress RBridges. Ingress-to-egress options MAY also be examined and acted upon by transit RBridges as specified in the particular option.

The second highest order bit of the first octet (NC) is zero for critical options and one for non-critical options.

Bit 10 in the second octet (MT) is zero for immutable options and one for mutable options. The IE, NC, Type, and MT fields themselves MUST NOT be changed even for a mutable option.

The eight-bit Type code extends from bit 2 through bit 9. The option Type may constrain the values of the IE, NC, and MT bits. For example, a certain Type might require that the option be marked as a hop-by-hop, non-critical, mutable option. If the IE, NC, or MT bits have a value not permitted by the option Type specification for an option that an RBridge must act on (any critical ingress-to-egress option at an egress RBridge and any critical hop-by-hop option), the RBridge MUST discard the frame. If these bits have a value not permitted by for the Type for an option that an RBridge may ignore (any ingress-to-egress option at a transit RBridge and any non-critical option), the RBridge MAY discard the frame. "MAY" is chosen in this case to minimize the checking burden.

The Length field is an unsigned quantity giving the length of the option value in units of four octets. It gives the size of the option including the initial two Type and Length octets. The Length field MUST NOT be such that the option value extends beyond the end of the total options area as specified by the TRILL Header Op-Length. Thus, the value 31 is reserved and, when such a value is noticed in a frame, the frame MUST be discarded.

2.3.3 Marshaling of Options

In a TRILL Header with options, those options start immediately after the Ingress RBridge Nickname and fill the options area. TLV options are 32-bit aligned.

TLV options start immediately after the initial four or eight octets of extended flags area and MUST appear in ascending order by the value of the eleven high order bits (bits 0 through 10) of the Type and Length octets considered as an unsigned integer in network byte order. There MUST NOT be more than one option in a frame with any particular value of this eleven high order bits. Thus the TLV options MUST be ordered as follows: (1) critical hop-by-hop options, (2) non-critical hop-by-hop options, (3) critical ingress-to-egress options, and (4) non-critical ingress-to-egress options. Frames that violate this paragraph are erroneous, will produce unspecified results, and MAY be discarded. "MAY" is chosen to minimize the format-checking burden on transit RBridges.

If any options are present, those options, both flag and TLV, MUST be correctly summarized into the CHbH, CItE, and TLV summary bits.

2.4 Conflict of Options

It is possible for options to conflict. Two or more options can be present in a frame that direct an RBridge processing the frame to do conflicting things or to change its interpretation of later parts of the frame in conflicting ways. Such conflicts are resolved by applying the following rules in the order given:

1. Any frame containing options that require mutually incompatible

changes in way later parts of the frame, after the options area, are interpreted or structured MUST be discarded. (Such options will be critical options, normally hop-by-hop critical options.)

2. Critical options override non-critical options.

2. Within each of the two categories of critical and non-critical options, the option appearing first in lexical order in the frame always overrides an option appearing later in the frame. Thus a conflict between an extended flag and a TLV option is always resolved in favor of the extended flag. Extended flags with lower bit numbers are considered to have occurred before extended flags with higher bit numbers.

3. Specific Extended Header Flag

The table below shows the state of TRILL Header extended flag assignments and the location of the special Flow ID field. See Section 6 for IANA Considerations.

Bits	Purpose	Section
0-1	Critical Summary Bits	2.3
2	More extended flags	2.
3-4	available for critical hop-by-hop flags	
5	available for non-critical hop-by-hop flag	
6-7	ECN	3.1
8-10	available for critical ingress-to-egress flags	
11-12	available for non-critical ingress-to-egress flags	
13-15	TLV Summary Bits	2.3.1.4
16-31	Flow ID	
32-39	available for critical hop-by-hop flags	
40-47	available for non-critical hop-by-hop flags	
48-55	available for critical ingress-to-egress flags	
56-63	available for non-critical ingress-to-egress flags	

Table 1. Extended Flag Options

3.1 The ECN Option

RBridges MAY implement an ECN (Explicit Congestion Notification) option [RFC3168]. If implemented, it SHOULD be enabled by default but can be disable on a per RBridge basis by configuration.

RBridges that do not implement this option or on which it is disabled simply (1) set bits 6 and 7 of the extended flags area to zero when they add an options area to a TRILL Header and (2) transparently copy those bits, if an options area is present, when they forward a frame with a TRILL Header.

An RBridge that implements the ECN option does the following, which correspond to the recommended provisions of [RFC6040], when that option is enabled:

- o When ingressing an IP frame that is ECN enabled (non-zero ECN field), it MUST add an options area to the TRILL Header and copy the two ECN bits from the IP header into extended header flags 6 and 7.
- o When ingressing a frame for a non-IP protocol, where that protocol has a means of indicating ECN that is understood by the RBridge, it MAY add an options area to the TRILL Header with the ECN bits set from the ingressed frame.

- o When forwarding a frame encountering congestion at an RBridge, if an options area is present with extended flags 6 and 7 indicating ECN-capable transport, the RBridge MUST modify them to the congestion experienced value.
- o When egressing an IP frame, the RBridge MUST set the outgoing native IP frame ECN field to the codepoint at the intersection of the values for that field in the encapsulated IP frame (row) and the TRILL extended Header ECN field (column) in Table 3 below or drop the frame in the case where the TRILL header indicates congestion experienced but the encapsulated native IP frame indicates a not ECN-capable transport. (Such frame dropping is necessary because IP transport that is not ECN-capable requires dropped frames to sense congestion.)
- o When egressing a non-IP protocol frame with a means of indicating ECN that is understood by the RBridge, it MAY set the ECN information in the egressed native frame by combining that information in the TRILL extended header and the encapsulated non-IP native frame as specified in Table 3.

The following table is modified from [RFC3168] and shows the meaning of bit values in TRILL Header extended flags 6 and 7, bits 6 and 7 in the IPv4 TOS Byte, and bits 6 and 7 in the IPv6 Traffic Class Octet:

Binary	Meaning
00	Not-ECT (Not ECN-Capable Transport)
01	ECT(1) (ECN-Capable Transport(1))
10	ECT(0) (ECN-Capable Transport(0))
11	CE (Congestion Experienced)

Table 2. ECN Field Bit Combinations

Table 3 below (adapted from [RFC6040]) shows how, at egress, to combine the ECN information in the extended TRILL Header ECN field with the ECN information in an encapsulated frame to produce the ECN information to be carried in the resulting native frame.

Inner Native Header	Arriving TRILL Header ECN Field			
	Not-ECT	ECT(0)	ECT(1)	CE
Not-ECT	Not-ECT	Not-ECT(*)	Not-ECT(*)	<drop>(*)
ECT(0)	ECT(0)	ECT(0)	ECT(1)	CE
ECT(1)	ECT(1)	ECT(1)(*)	ECT(1)	CE
CE	CE	CE	CE(*)	CE

Table 3: Egress ECN Behavior

An RBridge detects congestion either by monitoring its own queue depths or from participation in a link-specific protocol. An RBridge implementing the ECN option MAY be configured to add congestion experienced marking using ECN to any frame with a TRILL Header that encounters congestion even if the frame was not previously marked as ECN-capable or did not have an options area.

4. Specific TLV Option

The table below shows the state of TRILL Header TLV option Type assignment. See Section 6 for IANA Considerations.

Type	Purpose	Section
0x00	reserved	
0x00-0x7F	available	
0x80	Test/Pad	4.1
0x81-0xFE	available	
0xFF	reserved	

Table 4. TLV Option Types

The following subsection specifies a particular TRILL TLV option.

4.1 Test/Pad Option

This option is intended for testing and padding.

A specific meaning for this option with the critical flag set will not be defined so, in that form, it MUST always be treated as an unknown critical option. If the critical flag is not set, the option does nothing. In either case, it may be any length that will fit. Thus, for example, in the non-critical form, it can be used to cause the encapsulated frame staring right after the options area to be 64-bit aligned or for testing purposes.

- o Type is 0x80.
- o Length is variable. The value is ignored.
- o IE may be zero or one. This option has both hop-by-hop and ingress-to-egress versions.
- o NC is zero for the pad option and one for the test option.
  - \+ The non-critical version of this option does nothing.
  - \+ The critical version of this option MUST always be treated as an unknown critical option.
- o MT may be zero or one except that it must be zero if the other flags indicate the options is a critical hop-by-hop option. This option may be flagged as mutable or immutable.

5. Additions to IS-IS

RBridges use IS-IS PDUs to inform other RBridges which options they support. The specific IS-IS PDUs, TLVs, or sub-TLVs used to encode and advertise this information are specified in a separate document. Support for critical options MUST be advertised. Support for non-critical options MAY be advertised unless the specification of a particular non-critical option imposes a requirement higher than "MAY" for the advertising of that option by RBridges that implement it.

6. IANA Considerations

IANA will create two subregistries within the TRILL registry. A "TRILL Extended Header Flags" subregistry that is initially populated as specified in Table 1 in Section 3. And a "TRILL TLV Option Types" subregistry that is initially populated as specified in Table 4 in Section 4. References in both of those tables to sections of this document are to be replaced in the IANA subregistries by references to this document as an RFC.

New TRILL bit options and TLV option types are allocated by IETF Review [RFC5226].

7. Security Considerations

For general TRILL protocol security considerations, see [RFCtrill].

In order to facilitate authentication, options SHOULD be specified so they do not have alternative equivalent forms. Authentication of anything with alternative equivalent forms almost always requires canonicalization that an authenticating RBridge ignorant of the option would be unable to do and that may be complex and error prone even for an RBridge knowledgeable of the option. It is best for any option to have a unique encoding.

8. Acknowledgements

The following are thanked for their contributions: Bob Briscoe.

9. References

Normative and informative references for this document are given below.

9.1 Normative References

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

[RFC3168] - Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC5226] - Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC6040] - Briscoe, B., "Tunneling of Explicit Congestion Notification", RFC 6040, November 2010

[RFCtrill] - Perlman, R., D. Eastlake, D. Dutt, S. Gai, and A. Ghanwani, "RBridges: Base Protocol Specification", draft-ietf-trill-rbridge-protocol-16.txt, in RFC Editor’s queue.

9.2 Informative References

None.

Change History

The sections below summarize changes between successive versions of this draft. RFC Editor: Please delete this section before publication.

Version 00 to 02

Change the requirement for TLV option ordering to be strictly ordered by the value of the top nine bits of their first two bytes so that the MT bit is included.

Specify meaning of mutability bit for hop-by-hop options.

Fix length of Flow ID Value at 2.

Require that options that may significantly affect the interpretation or format of subsequent parts of the frame be critical options.

Version 02 to 03

Move Test/Pad option into this document from the More Options draft and move the More Flags option from this document into the More Options draft.

Prohibit multiple occurrences of a TLV option in a frame.

Version 03 to 04

Restructure the bit encoded options area so that the initial 32 bits include a 16 bit Flow ID, various TLV-option-present bits, and a more extended flags bit that means another 32 bits of extended flags are present.

Change the Length of TLV encoded options so that it is in units of 4 bytes, not 1, resulting in a bigger Type field.

Update Explicit Congestion Notification to follow RFC 6040.

Rename "bit encoded options" to be "extended header flags" or "extended flags".

Authors' Addresses

Donald Eastlake
Huawei Technologies
155 Beaver Street
Milford, MA 01757

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

Anoop Ghanwani
Brocade Communications Systems
130 Holger Way
San Jose, CA 95134 USA

Phone: +1-408-333-7149
Email: anoop@brocade.com

Vishwas Manral
IP Infusion Inc.
1188 E. Arques Ave.
Sunnyvale, CA 94089 USA

Tel: +1-408-400-1900
email: vishwas@ipinfusion.com

Caitlin Bestler
Quantum
1650 Technology Drive , Suite 700
San Jose, CA 95110

Phone: +1-408-944-4000
email: cait@asomi.com

Copyright and IPR Provisions

Network Working Group V. Manral, Ed.
Internet-Draft IPInfusion Inc.
Intended status: Standards Track D. Eastlake
Expires: September 14, 2011 Huawei Inc.
D. Ward
Juniper Networks
A. Banerjee
Cisco Systems
March 13, 2011

Rbridges: Bidirectional Forwarding Detection (BFD) support for TRILL
draft-manral-trill-bfd-encaps-01

Abstract

This document specifies use of the BFD (Bidirectional Forwarding Detection) protocol in RBridge campuses based on the OAM (Operations, Administration, and Maintenance) Channel extension to the TRILL (TRansparent Interconnection of Lots of Links) protocol.

BFD is a widely deployed OAM mechanism in IP and MPLS networks. However, in the present form a BFD packet cannot be sent over a TRILL network as it is either IP/ UDP encapsulated or encapsulated directly over MPLS or using ACH encapsulation. This document also defines BFD encapsulation over TRILL to address this shortcoming.

Requirements Language

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

This Internet-Draft will expire on September 14, 2011.

Copyright Notice

Table of Contents

1. Introduction

Faster convergence is a very critical feature of TRILL networks. The TRILL IS-IS Hellos used between RBridges provide a basic neighbor and continuity check for TRILL links. However, failure detection by non-receipt of such Hellos is based on the holding time parameter which is commonly set to a value of tens of seconds and, in any case, has a minimum expressible value of one second.

Some applications, including voice over IP, may wish, with high probability, to detect interruptions in continuity within a much shorter time period. In some cases physical layer failures can be detected very rapidly but this is not always possible, such as when there is a failure between two bridges that are in turn between two RBridges. There are also many subtle failures possible at higher levels. For example, some forms of failure could affect unicast frames while still letting multicast frames through; since all TRILL IS-IS Hellos are multicast such a failure cannot be detected with Hellos. Thus, a low overhead method for frequently testing continuity for the TRILL Data between neighbor RBridges is necessary for some applications. BFD protocol provides a low-overhead, short-duration detection of failures in the path between forwarding engines.

This document describes a TRILL encapsulation for BFD packets for networks that do not use IP addressing or for ones where it is not desireable.

2. Terminology

BFD: Bi-directional Forwarding Detection

OAM: Operations, Administration, and Maintenance

MPLS: Multi Protocol Label Switching

IS-IS: Intermediate-System to Intermediate-System

TTL: Time To Live

3. BFD over TRILL

TRILL supports neighbor BFD Echo and one-hop and multi-hop BFD Control, as specified below, over the TRILL OAM Channel facility. Multi-destination BFD is beyond the scope of this document. The OAM Channel facility is specified in [TRILLoam].

BFD over TRILL support is similar to BFD over IP support except where it is explicitly so mentioned. When running BFD over TRILL both Single Hop as well as in Multi Hop sessions are supported.

Asynchronous mode is supported, however the demand mode is not supported for TRILL. BFD over TRILL supports the Echo function, however this can be used for only Single hop sessions.

The TRILL Header Hop count in the BFD packets sent out with a value of 63. To prevent spoofing attacks, the TRILL Hop count of a received session is checked. For a single Hop session if the Hop count is less than 63 the packet is discarded if the GTSM mode [RFC5082] is set. For Multi Hop sessions the Hop count check can be disabled or the bfdTrillAcceptedHopCount value can be configured. If a packet is received with a hop count of less than bfdTrillAcceptedHopCount, the packet is discarded.

The format of the echo packet is not defined.

A new BFD TRILL header is defined.

Authentication mechanisms as supported in BFD are also supported for BFD running over TRILL.

4. Sessions and Initialization

Within an RBridge campus, there will be only a single TRILL BFD Control session between two RBridges over a given interface visible to TRILL. This BFD session must be bound to this interface. As such, both sides of a session MUST take the "Active" role (sending initial BFD Control packets with a zero value of Your Discriminator), and any BFD packet from the remote machine with a zero value of Your Discriminator MUST be associated with the session bound to the remote system and interface.

Note that TRILL BFD provides OAM facilities for the TRILL Data plane. This is above whatever protocol is in use on a particular link, such as a PPP [TrillPPP] link or an Ethernet link. Link technology specific OAM protocols may be used on a link between neighbor RBridges, for example Continuity Fault Management [802.1ag] if the link is Ethernet. But such link layer OAM and coordination between it and TRILL data plaen layer OAM, such as TRILL BFD, is beyond the scope of this document.

If lower level mechanisms, such as link aggregation [802.1AX], are in use that present a single logical interface to TRILL IS-IS, only a single TRILL BFD session can be established to any other RBridge over this logical interface. However, lower layer OAM could be aware of and/or run separately on each of the components of an aggregation.

5. Relationship to MPLS OAM

TRILL BFD uses the TRILL OAM Channel [TRILLoam] is the same way that MPLS OAM protocols use the MPLS Generic Associated Channel [RFC5586]. However, the RBridges that implement TRILL are IS-IS based routers, not label switched routers; thus TRILL BFD is closer to IPv4/IPv6 BFD than to MPLS BFD.

TRILL BFD optionally includes support of BFD Echo which is not specified for MPLS BFD.

6. TRILL BFD Control Protocol

TRILL BFD Control frames are unicast TRILL OAM Message Channel frames [TRILLoam]. The TRILL OAM Protocol value is given in Section 4.

The protocol specific data associated with the TRILL BFD Control protocol is as shown below. See [RFC5880] for further information on these fields.

```
 TRILL BFD Control Protocol Data:
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Vers |  Diag   |Sta|P|F|C|A|D|M|  Detect Mult  |    Length     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       My Discriminator                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Your Discriminator                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    Desired Min TX Interval                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Required Min RX Interval                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Required Min Echo RX Interval                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 Optional Authentication Section:
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Auth Type   |   Auth Len    |    Authentication Data...     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

7. One-Hop TRILL BFD Control

One-hop TRILL BFD Control is typically used to rapidly detect link and RBridge failures. TRILL BFD frames over one hop for such purposes SHOULD be sent with priority 7.

For neighbor RBridges RB1 and RB2, each RBridge sends one-hop TRILL BFD Control frames to the other only if TRILL IS-IS has detected bi-directional connectivity and both RBridges indicate support of TRILL BFD is enabled. The BFD Enabled TLV is used to indicate this as specified in [RFCbfdtlv]. The indication of TRILL BFD support with the BFD Enabled TLV overrides any indication of lack of support through failure to indicate support of the OAM-Channel TRILL Header extended flag.

8. BFD Control Frame Processing

The following tests SHOULD be performed on received TRILL BFD Control frames before generic BFD processing.

Is the M bit in the TRILL Header non-zero? If so, discard the frame. TRILL support of multi-destination BFD Control is beyond the scope of this document.

If the OAM Header MH flag is zero, indicating one-hop, test that the TRILL Header hop count received was 0x3F (i.e., is 0x3E if it has already been decremented) and if it is any other value discard the frame. If the MH OAM flag is one, indicating multi-hop, test that the TRILL Header hop count received was not less than a configurable value that defaults to 0x30. If it is less, discard the frame.

9. TRILL BFD Echo Protocol

A TRILL BFD Echo frame is a unicast TRILL OAM Message Channel frame, as specified in [TRILLoam], which should be bounced back by an immediate neighbor because both the ingress and egress nicknames are set to a nickname of the originating RBridge. Normal TRILL Data frame forwarding will cause the frame to be returned. The TRILL OAM protocol number for BFD Echo is given in Section 4.

TRILL BFD Echo frames SHOULD only be sent on a link if

A TRILL BFD Control session has been established,

TRILL BFD Echo support is indicated by the potentially echo responding RBridge, and

The TRILL BFD Echo originating RBridge wishes to make use of this optional feature.

Since the originating RBridge is the RBridge that will be processing a returned Echo frame, the entire TRILL BFD Echo protocol specific data area is considered opaque and left to the discretion of the originating RBridge. Nevertheless, it is RECOMMENDED that this data include information by which the originating RBridge can authenticate the returned BFD Echo frame and confirm the neighbor that echoed the frame back. For example, it could include its own SystemID, the neighbor’s SystemID, a session identifier and a sequence count as well as a Message Authentication Code.

9.1. BFD Echo Frame Processing

The following tests SHOULD be performed on returned TRILL BFD Echo frames before other processing. (In some implementations, the TRILL Header may not be available to the TRILL BFD Echo module in which case these check are not possible.)

Is the M bit in the TRILL Header non-zero? If so, discard the frame. TRILL support of multi-destination BFD Echo is beyond the scope of this document.

The TRILL BFD Echo frame should have gone exactly two hops so test that the TRILL Header hop count as received was 0x3E (i.e., 0x3D if it has already been decremented) and if it is any other value discard the frame. The TRILL OAM Header in the frame should have the MH bit equal to one and if it is zero, the frame is discarded.

10. Management and Operations Considerations

The TRILL BFD parameters at an RBridge are configurable... The default values are ... TBD.

It is required that the operator of an RBridge campus configure the rates at which TRILL BFD frames are transmitted on a link to avoid congestion (e.g., link, I/O, CPU) and false failure detection.

11. Security Considerations

This draft raises no new security considerations than those already mentioned in the BFD [RFC5880]. By keeping a seperate flag for Single Hop and Multihop sessions it allows the TTL check to be performed thus preventing spoofing of packets.

However the same is possible even without the changes mentioned in this document. A device should rate limit the LSP ping packets redirected to the CPU so that the CPU is not overwhelmed.

12. IANA Considerations

IANA is request to allocate two TRILL OAM Protocol numbers from the range allocated by Standards Actions, as follows:

Protocol	Number
BFD Control	TBD (2 suggested)
BFD Echo	TBD (3 suggested)

13. Acknowledgements

The authors would like to thank a lot of folks. Names will be disclosed soon.

14. References

14.1. Normative References

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

[RFC4634] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC 4634, July 2006.

[RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.

[RFC5586] Bocci, M., Vigoureux, M., and S. Bryant, "MPLS Generic Associated Channel", RFC 5586, June 2009.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

14.2. Informative References

[802.1AX] IEEE, "IEEE Standard for Local and metropolitan area networks / Link Aggregation", 802.1AX-2008, 1 January 2008.

[802.1ag] IEEE, "IEEE Standard for Local and metropolitan area networks / Virtual Bridged Local Area Networks / Connectivity Faul
t
Management", 802.1ag-2007, 17 December 2007.

[TrillPPP] Carlson, J., "PPP TRILL Protocol Control Protocol", draft-ietf-pppext-trill-protocol-02.txt, work in progress, May 201
0.

Authors' Addresses

Vishwas Manral (editor)
IPInfusion Inc.
1188 E. Arques Ave.
Sunnyvale, CA 94085
USA

Phone: 408-400-1900
Email: vishwas@ipinfusion.com

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

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

Dave Ward
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089-1206
USA

Phone: 408-745-2000
Email: dward@juniper.net

Ayan Banerjee
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95138
USA

Phone: 408-525-8781
Email: ayabaner@cisco.com

TRILL Working Group   Radia Perlman
INTERNET-DRAFT   Intel Labs
Intended status: Informational   Donald Eastlake
                                                                  Huawei
                                                          Anoop Ghanwani
                                                                 Brocade
Expires: September 6, 2011   March 7, 2011

RBridges: Multilevel TRILL
<draft-perlman-trill-rbridge-multilevel-01.txt>

Abstract

This document describes issues, and various possible approaches, to extending TRILL to use multiple levels of IS-IS.

Status of This Memo

Acknowledgements

The helpful comments of the following are hereby acknowledged: David Michael Bond.

Table of Contents

1. Introduction

The IETF TRILL protocol [RFCtrill] [RFCadj] 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] link state routing and encapsulating traffic using a header that includes a hop count. The design supports VLANs and optimization of the distribution of multi-destination frames based on VLANs and IP derived multicast groups. Devices that implement TRILL are called RBridges.

Familiarity with [RFCtrill] is assumed in this document.

1.1 TRILL Scalability Issues

There are multiple issues that might limit the scalability of a TRILL-based network:

- o the routing computation load,
- o the volatility of the LSP database creating too much control traffic,
- o the volatility of the LSP database causing the TRILL network to be in an unconverged state too much of the time,
- o the size of the LSP database,
- o the size of the end node learning table (the table that remembers (egress RBridge, VLAN/MAC) pairs),
- o the traffic due to upper layer protocols use of broadcast and multicast, and
- o the hard limit of the number of RBridges, due to the 16-bit nickname space.

Extending TRILL IS-IS to be multilevel (hierarchical) helps with some of these issues.

IS-IS was designed to be multilevel [IS-IS] [RFC1195] be partitioned into "areas". Routing within an area is known as "level 1 routing". Routing between areas is known as "level 2 routing". The level 2 IS-IS network consists of level 2 routers and links between the level 2 routers. Level 2 routers may participate in one or more areas, in addition to their role as level 2 routers.

Each area is connected to the level 2 area through one or more "border routers", which participate both as a router inside the area, and as a router inside the level 2 "area".

1.2 Improvements Due to Multilevel

Partitioning the network into areas reduces the size of the LSP database in each router, and stops volatility of the topology in one area from disrupting other areas. Allowing TRILL to utilize IS-IS’s hierarchy solves the first 4 issues above, but does not necessarily help the other 3 issues (size of end node learning table, traffic due to upper layer protocols using multicast, hard limit of 16-bit RBridge nicknames).

We propose two variants of hierarchical or multilevel TRILL. One we call the "unique nickname" variant. The other we call the "aggregated nickname" variant. In the aggregated nickname variant, border RBridges replace either the ingress or egress nickname field in the TRILL header of unicaat packets with an aggregated nickname representing an entire area.

The aggregated nickname variant has the following advantages:
- o it solves the 16-bit RBridge nickname limit,
- o it lessens the amount of inter-area routing information that must be passed in IS-IS,
- o it greatly reduces the RPF information (since only the area nickname needs to appear, rather than all the ingress RBridges in that area), and
- o it enables computation of trees such that the portion computed within a given area is rooted within that area.

The unique nickname variant has the advantage that border RBridges do not need to do end node learning for end nodes in their own area.

1.3 More on Areas

Each area is configured with an "area address", which is advertised in IS-IS messages, so as to avoid accidentally interconnecting areas. Note that although the area address had other purposes in CLNP, (IS-IS was originally designed for CLNP/DECnet), for TRILL the only purpose of the area address would be to avoid accidentally interconnecting areas.

Currently, the TRILL specification says that the area address "must be zero". If we change the specification so that the area address value of zero is a default, then most of IS-IS multilevel machinery works as originally designed. However, there are some TRILL-specific issues, which we address below in this document.

1.4 Terminology and Acronyms

This document uses the acronyms defined in [RFCtrill] and the following additional acronym:

DBRB - Designated Border RBridge

2. Multilevel TRILL Issues

The TRILL-specific issues introduced by hierarchy include the following:

a) configuration of non-zero area addresses, encoding them in IS-IS PDUs, and interworking with old RBridges that do not understand nonzero area addresses,
b) nickname management,
c) advertisement of filtering information (VLAN reachability, IP multicast addresses) across areas,
d) computation of trees across areas for multi-destination frames,
e) computation of RPF information for those trees, and
g) compatibility, as much as practical, with existing, unmodified RBridges. The most important form of compatibility is with existing TRILL fast path hardware. Changes that require upgrade to the slow path firmware/software are more tolerable.

Filtering information is only an optimization, as long as multidestination frames are not prematurely filtered. Thus, for instance, border RBridges could advertise they can reach all possible VLANs, and have an IP multicast router attached. This would cause multidestination traffic to be transmitted to the border router, and possibly filtered there, when the traffic could have been filtered earlier based on VLAN or multicast group.

2.1 Non-zero Area Addresses

The current TRILL base protocol specification [RFCtrill] says that the area address in IS-IS MUST be zero. The purpose of the area address is to ensure that different areas are not accidentally hooked together. Furthermore, zero is an invalid area address for layer 3 IS-IS, so it was chosen as an additional safety mechanism to ensure that layer 3 IS-IS would not be confused with TRILL IS-IS. However, TRILL uses a different multicast address and Ethertype to avoid such confusion, so it is not necessary to worry about this.

Since current TRILL RBridges will reject any IS-IS messages with nonzero area addresses, the choices are; all RBridges must be upgraded, neighbors of old RBridges must remove the area address from IS-IS messages when talking to an old RBridge (which might cause inadvertent merging of areas), to ignore the problem of accidentally merging areas entirely, or to keep the fixed "area address" field as 0 in TRILL, and add a new, optional TLV for "area name" that, if present, could be compared, by new RBridges, to prevent accidental merging

2.2 Aggregated versus Unique Nicknames

In the unique nickname variant, all nicknames across the campus must be unique. In the aggregated nickname variant, RBridge nicknames are only of local significance within an area, and the only nickname externally (outside the area) visible is the "area nickname", which aggregates all the internal nicknames.

The aggregated nickname approach eliminates the potential problem of nickname exhaustion, minimizes the amount of nickname information that would need to be forwarded between areas, minimizes the size of the forwarding table, and simplifies RPF calculation and RPF information.

With unique cross-area nicknames, it would be intractable to have a flat nickname space with RBridges in different areas contending for the same nicknames. Instead, each area would need to be configured with a block of nicknames. Either some RBridges would need to announce that all the nicknames other than that block are taken (to prevent the RBridges inside the area from choosing nicknames outside the area’s nickname block), or a new TLV would be needed to announce the allowable nicknames, and all RBridges in the area would need to understand that new TLV.

Currently the encoding of nickname information in TLVs does not allow any aggregation. The information could be encoded as ranges of nicknames to make this somewhat manageable; however, a new TLV for announcing nickname ranges would not be intelligible to old RBridges.

In contrast, the aggregated nickname approach enables passing far less nickname information and works as follows:

Each area would be assigned a 16-bit nickname. This would not be the nickname of any actual RBridge. Instead, it would be the nickname of the area itself. Border RBridges would know the area nickname for their own area(s).

In the following picture, R2 and R3 are area border RBridges. A source S is attached to R1. The two areas have nicknames 15961 and 15918, respectively. R1 has a nickname, say 27, and R4 has a nickname, say 44 (and in fact, they could even have the same nickname, since the RBridge nickname will not be visible outside the area).

```
             Area 15961               level 2            Area 15918
     +-------------------+     +-----------------+    +-------------+
     |                   |     |                 |    |             |
     |  S--R1---Rx--Rz-----R2----Rb---Rc--Rd---Re--R3---Rk--R4---D  |
     |     27            |     |                 |    |     44      |
     |                   |     |                 |    |             |
     +-------------------+     +-----------------+    +-------------+
```

Let's say that S transmits a packet to destination D, and let's say that D's location is learned by the relevant RBridges already. The relevant RBridges have learned the following:

1) R1 has learned that D is connected to nickname 15918
2) R3 has learned that D is attached to nickname 44.

The following sequence of events will occur:

- S transmits an Ethernet packet with source MAC = S and destination MAC = D.

- R1 encapsulates with a TRILL header with ingress RBridge = 27, and egress = 15918.

- R2 has announced in the level 1 IS-IS instance in area 16961, that it is attached to all the area nicknames, including 15918. Therefore, IS-IS routes the packet to R2. (Alternatively, if a distinguished range of nicknames is used for area, Level 1 RBridges seeing such an egress nickname will know to route to the nearest border router.)

- R2, when transitioning the packet from level 1 to level 2, replaces the ingress RBridge nickname with the area nickname, so replaces 27 with 15961. Within level 2, the ingress RBridge field in the TRILL header will therefore be 15961, and the egress RBridge field will be 15918. Also R2 learns that S is attached to nickname 27 in area 15961.

- The packet is forwarded through level 2, to R3, which has advertised, in Level 2, reachability to the nickname 15918.

- R3, when forwarding into area 15918, replaces the egress nickname in the TRILL header with R4's nickname (44). So, within the destination area, the ingress nickname will be 15961 and the egress nickname will be 44.

- R4, when decapsulating, learns that S is attached to nickname 15961.

Now suppose that D's location has not been learned by R1 and/or R3. What will happen, as it would in TRILL today, is that R1 will forward the packet as a multidestination frame, choosing a tree. As the multidestination frame transitions into level 2, R2 replaces the ingress nickname with the area nickname.

Now suppose that R1 has learned the location of D (attached to nickname 15918), but R3 does not know where D is. In that case, R3 will turn the packet into a multidestination frame within the area. Care must be taken so that, in case R3 is not the Designated transitioner for that multidestination frame, but was on the unicast path, that another RBridge within that area not forward the now multidestination frame back into level 2. Therefore, it would be desirable to have a marking, somehow, that indicates the scope of this packet to be "only this area".

There is an issue with tree nicknames that would be a problem with the unique nickname variant, but is solved with the aggregated variant, as follows:

Suppose nicknames were unique within the TRILL campus, and that the TRILL header was not rewritten by the border RBridges. In that case, there would have to be globally known nicknames for the trees. Suppose there are k trees. For all of the trees with nicknames located outside an area, the trees would all be rooted at (one of) the border RBridge(s). Therefore, there would be no path splitting of multidestination with the area.

In contrast, with the aggregated nickname solution, each border RBridge can have a mapping from the level 2 tree nickname to the level 1 tree nickname. There need not even be agreement about the total number of trees; just that the border RBridge have some mapping, and replace the egress RBridge nickname (the tree name) when transitioning levels.

Care must be taken that it be clear, when transitioning between level 2 and area X, which (single) border RBridge will transition the packet between the levels.

2.3 Building Multi-Area Trees

It is easy to build a multi-area tree by building a tree in each area separately, (including the level 2 "area"), and then having only a single border RBridge, say R1, in each area, attach to the level 2 area. R1 would forward all multidestination packets between that area and level 2.

People might find this unacceptable, however, because of the desire to path split (not always sending all multidestination traffic through the same border RBridge).

Having multiple border RBridges introduces some complexity:

a) calculating the RPF check when a multidestination frame originates outside the area (which border RBridge injected the frame into the area?)

b) calculating the filtering information (which border RBridge will transition the frame into level 2?)

This might be solvable if all RBridges are multilevel aware, however it is difficult to imagine how to ensure that old RBridges would calculate RPF and filtering information sensibly.

Ignoring old RBridges for now, various possible solutions are

a) elect one border RBridge for transitioning all multidestination frames between levels (call that the Designated Border RBridge (DBRB))

b) allow the DBRB to appoint other border RBs to forward some subset of the inter-level frames. (as the DRB does, on a per-VLAN basis, on a link). Make the appointment information visible to the other RBridges in the area so that they can calculate their RPF and filtering information.

If b), then on what basis would the appointment be made? Various possibilities are as follows:

- o based on VLAN
- o based on tree root
- o based on ingress RBridge nickname

The more flexibility that is allowed, the more complex announcement of information becomes, and the more complex the tree database becomes. If appointment is made based on VLAN, then the RPF check would need to be based on (tree, VLAN, ingress nickname), rather than simply (tree, ingress nickname) as it is today.

2.4 The RPF Check for Trees

For multidestination frames originating in R1’s area, computation of the RPF check is done as today. For multidestination frames originating outside R1’s area, computation of the RPF check must be done based on one of the border RBridges (say R1, R2, or R3).

An RBridge, say R4, located inside an area, must be able to know which of R1, R2, or R3 transitioned the frame into the area from level 2. (or into level 2 from an area).

This could be done based on having the DBRB announce the assignments to all the RBs in the area.

2.5 Area Nickname Acquisition

In the aggregated nickname variant, each area must acquire a unique area nickname. It is probably simpler to allocate a block of nicknames (say, the top 2000) to be area addresses, and not used by any RBridges.

The area nicknames need to be advertised and acquired through level 2.

Within an area, all the border RBridges must discover each other through the level 1 IS-IS database, by advertising, in their LSP "I am a border RBridge".

Of the border RBridges, one will have highest priority (say R7). It will be R7 that dynamically participates, in level 2, to acquire a nickname for the area. R7 will give the area a pseudonode name, such as R7.5, within level 2. So an area will appear, in level 2, as a pseudonode.

The pseudonode will participate, in level 2, in acquiring a nickname for the area.

Within level 2, all the border RBridges [for the area] advertise reachability to the pseudonode, which will mean connectivity to the area nickname.

2.6 Link state representation of areas

Within an area, say area A, there is an election for the DBRB, (Designated Border RB), say R1. This will be done through LSPs within area A. The border RBs announce themselves, together with DBRB priority. (Note that the election of the DBRB cannot be done based on Hello messages, because the border RBs are not necessarily physical neighbors of each other. They can, however, reach each other through connectivity within the area, which is why it will work to find each other through level 1 LSPs.)

R1 acquires the area nickname (in the aggregated nickname approach), gives the area a pseudonode name (just like the DRB would give a pseudonode name to a link). R1 advertises, in area A, what the pseudonode name for the area is (and the area nickname that R1 has acquired).

The pseudonode LSP initiated by R1 includes any information extraneous to area A that should be input into area A (such as area nicknames of external areas, or perhaps (in the unique nickname variant), all the nicknames of external RBs in the TRILL campus and filtering information such as IP multicast groups and VLANs). All the other border RBs for the area announce (in their LSP) attachment to that pseudonode.

Within level 2, R1 generates a level 2 LSP on behalf of the area, also represented as a pseudonode. The same pseudonode name could be used within level 1 and level 2, for the area. (There does not seem any reason why it would be useful for it to be different, but there's also no reason why it would need to be the same). Likewise, all the area A border RBs would announce, in their level 2 LSPs, connection to the pseudonode.

3. Area Partition

It is possible for an area to become partitioned, so that there is still a path from one section of the area to the other, but that path is via the level 2 area.

An area will naturally break into two areas in this case.

An area address might be configured to ensure two areas are not inadvertently connected. That area address appears in Hellos and LSPs within the area. If two chunks, connected only via level 2, were configured with the same area address, this would not cause any problems. (They would just operate as separate level 1 areas.)

A more serious problem occurs if the level 2 area is partitioned in such a way that it healed by using a path through a level 1 area. TRILL will not attempt to solve this problem. Within the level 1 area, a single border RBridge will be the DBRB, and will be in charge of deciding which (single) RBridge will transition any particular multidestination frames between that area and level 2. If the level 2 area is partitioned, this will result in multidestination frames only reaching the portion of the TRILL campus reachable through the partition attached to the RBridge that transitions that frame. It will not cause a loop.

4. Multidestination Scope

It would be desirable to be able to mark a multidestination frame with a scope that indicates this packet should not exit the area. This is particularly true when, in the aggregated nickname variant, a unicast packet turns into a multidestination packet.

This could be done by having two tree nicknames, for each tree; one being the tree "only for this area", and the other being for multi-area trees.

Alternatively, a packet intended only for the area could be tunneled (within the area) to the RBridge Rx, that is the appointed transitioner for that form of packet (say, based on VLAN), with instructions that Rx only transmit the packet within the area, and Rx could initiate the multidestination frame within the area. Since Rx introduced the frame, and is the only one allowed to transition that frame within levels, this would accomplish scoping of the packet to within the area.

Since this case would only occur when unicast frames need to be turned into multidestination (because the border RBridge in the destination area does not know the location of the destination), the suboptimality of tunneling between the border RBridge that receives the unicast frame and the appointed level transitioner for that frame, would not be an issue.

5. Co-Existence with Old RBridges

RBridges that are not multilevel aware have a problem with calculating RPF check and filtering information, since they would not be aware of assignment of border RBridge transitioning.

A possible solution, as long as any old RBridges exist within an area, is to have the border RBridges elect a single DBRB (Designated Border RBridge), and have all inter-area traffic go through the DBRB (unicast as well as multidestination). If that DBRB goes down, a new one will be elected, but at any one time, all inter-area traffic (unicast as well as multidestination) would go through that one DRBR.

6. Summary

This draft outlines the issues and possible approaches to multilevel TRILL. The variant involving area nicknames for aggregation has significant advantages in terms of scalability; not just of avoiding nickname exhaustion, but allowing, for instance, RPF checks to be aggregated based on an entire area.

Some issues are not difficult, such as dealing with partitioned areas. Some issues are more difficult, especially dealing with old RBridges.

7. Security Considerations

TBD

8. IANA Considerations

This document requires no IANA actions. RFC Editor: Please delete this section before publication.

9. References

Normative and Informational references for this document are listed below.

9.1 Normative References

[IS-IS] - ISO/IEC 10589:2002, Second Edition, "Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)", 2002.

[RFC1195] - Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[RFCtrill] - Perlman, R., D. Eastlake, D. Dutt, S. Gai, and A. Ghanwani, "RBridges: Base Protocol Specification", draft-ietf-trill-rbridge-protocol-16.txt, in RFC Editor’s queue.

[RFCadj] - Eastlake, D., R. Perlman, A. Ghanwani, D. Dutt, V. Manral, "RBridges: Adjacency", draft-ietf-trill-adj, work in progress.

9.2 Informative References

None.

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

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

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

Anoop Ghanwani
Brocade Communications Systems
130 Holger Way
San Jose, CA 95134 USA

Phone: +1-408-333-7149
Email: anoop@brocade.com

Copyright and IPR Provisions

Adaptive VLAN Assignment for Data Center RBridges
draft-zhang-trill-vlan-assign-00.txt

Abstract

When several RBridges are multi-accessed to a LAN link, each of them can act as the packet forwarder for the hosts attached to this link. One of the RBridges will be elected as the Designated RBridge (DRB) which is responsible to choose the Appointed Forwarder (AF) for each VLAN appearing on this LAN link. If the DRB casually assign a VLAN to an RBridge as the Appointed Forwarder without considering the number of the MAC addresses and traffic load of this VLAN, it may overload some of the RBridges while leave other RBridges lightly loaded. This unbalanced assignment issue reduces the scalability of a TRILL network and undermines its performance. Therefore, the TRILL DRB should choose Appointed Forwarders taking their load into consideration. The goal of this document is to design a new protocol to support the adaptive VLAN assignment (or Appointed Forwarder selection) based on the forwarders' reporting of their usage of MAC tables and available bandwidth.

Status of this Memo

Table of Contents

1 Introduction

The scales of Data Center Networks (DCNs) are expanding very fast these years. In DCNs, Ethernet switches and bridges are abundantly used for the interconnection of servers. The plug-and-play feature and the simple management and configuration of Ethernet are appealing to the DCN providers. A whole DCN can be a simple large layer 2 Ethernet which is either built on a real network or on a virtualization platform.

Cloud Computing is growing up from DCNs which can be seen as a virtualization platform that provides the reuse of the network resources of DCNs. A lot of cloud applications have been developed by DCN providers, such as Amazon’s Elastic Compute Cloud (EC2), Akamai’s Application Delivery Network (ADN) and Microsoft’s Azure. Cloud Computing clearly brings new challenges to the traditional Ethernet. The scales of the DCNs are becoming too large to be carried on the traditional Ethernet. The valuable MAC-tables of the bridges are running out of use for storing millions of MAC addresses. The broadcast of ARP messages consumes too much bandwidth and computing resources. The mobility of end stations brings dynamics to the network which can be a heavy burden if the management and configuration of the network involves too much manpower. The Spanning Tree Protocol used in the traditional Ethernet is outdated since there is only a single viable path on the tree for a node pair and this path is not always the best path (e.g., shortest path).

RBridges are designed to improve the shortcomings of the traditional Ethernet. To make use of the rich connections, RBridges introduce multi-pathing to the Ethernet to break the single-path constraint of STP. Multiple points of attachment is a basic feature supported by RBridges and common for Data Center Bridges. This feature not only increases the "east-west" capacity but also greatly enhances the reliability of DCNs [VL2] [SAN]. If several RBridges are attached to a bridged LAN link at the same time, the DRB is responsible for the assignment of a VLAN to one of the RBridges as the Appointed forwarder. However, the current VLAN assignment is done in a one-way manner. The DRB casually assign a VLAN to an RBridge attached to the local link without knowing its available MAC-table entries or bandwidth. The appointed forwarder does not feedback the utilization of its MAC-table or bandwidth either.

This document aims to open a feedback passageway from a appointed forwarder to its DRB. Two types of sub-TLVs are defined, with which a forwarder can report its MAC entries and traffic bit rate respectively. By gathering these report messages, the VLAN assignment can be done in a way that the usage of the MAC tables and bandwidth of the attached RBridges are balanced.

1.1 Terminology

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

2 Data Center RBridge

Data Center Networks grow rapidly recently. Ethernet is widely used in data centers because of its simple management and plug-and-play features. However, there are shortcomings of Ethernet. RBridges are designed to improve these shortcomings. In this section, we analyze the characteristics of the DCNs that impact the design of RBridges and reveal why the adaptive VLAN assignment is important for RBridges to be used in DCNs.

2.1 Scalability

In the past years, a large DCN is typically composed of tens of thousands servers interconnected through switches and bridges. In the future cloud computing era, there can be as many as millions of servers in one DCN. The management of the numerous MAC addresses of the servers on the layer2 devices will become more and more complex. RBridges are aimed to replace the traditional bridges. The valuable CAM-tables on RBridges can easily be used up if they are not used reasonably [CAMtable]. For RBridges to be widely used in DCNs, the VLANs should be assigned to the RBridges in a manner that the MAC entries of the VLANs on the RBridges are balanced.

2.2 East-West Capacity Increase

The Spanning Tree Protocol (STP) in the traditional LAN blocks some ports of the bridges for the purpose of loop avoidance. However, the side-effects of STP are obvious. The link bandwidth attached to the blocked ports are not used which greatly wastes the capacity of the network. On the tree topology, the communication between the bridges of the left branch and right branch must transit the single root bridge, which forms a "hair-pin turn".

With the rapid increase of the amount of servers in DCNs and their traffic demand, it is urgent to break the constraint of STP and enhance the "east-west" capacity of DCNs which are always richly connected. RBridges use the multi-path routing to set up the data plane of a TRILL network. Multiple RBridges may be attached to the same LAN link, which offers multiple access points to the LAN link. The hosts on this LAN link is therefore multi-homed to a TRILL network. All the attached RBridges can act as the packet forwarder for the VLANs carried on this LAN link. In the worst case, all the

VLANs are probably assigned to a single RBridge. Under this scenario, the ingress capacity on the other RBridges is wasted. It is necessary to balance the traffic load of the VLANs among these RBridges through the assignment of the VLANs.

2.3 Virtualization

Virtualization is important for increasing the utilization of network resources in DCNs. For example, the VPNs can be used to separate the traffic from different services therefore they can be carried on the same pool of resources. When the VPNs is carried over a TRILL network, RBridges can use a VLAN tag to identify each VPN. However, the use of VLANs multiplies the entries in the MAC table of the RBridges. Since a host can be a member of several VLANs at the same time, the RBridges have to store multiple copies of its MAC address in its precious MAC table.

Virtual Machines (VM) are widely used in DCNs. A physical host can support multiple VMs and each of the VMs has to be identified by one MAC address that is need to be stored in the MAC tables of the RBridges. This seriously increases the numbers of MAC entries in RBridges. Moreover, the number of VMs in a VLAN is not necessarily equal to the number of the physical hosts. VMs are spawned or destroyed based on the demand of the applications. They can also migrate from one location to another, which may be either an in-service or out-of-service move. VMs bring about the volatility of the size of VLANs. It is hard for a TRILL network to provide one static VLAN assignment based on the numbers of physical hosts of VLANs that is proper for all applications all the time. It is necessarily to do VLAN assignment adaptively.

3 MAC Entries Balancing

A CAM-table on a switch is expensive, which is a major constraint on the scalability of Ethernet [CAMtable]. When a RBridge is used to connect lots of hosts in large Data Center Networks, the entries of the CAM-table can easily be used up. The network should be tactically interconnected and the valuable MAC table entries should be used economically.

RBridges support multiple points of attachment [TRILLbase]. When RBridges are used in a DCN to form a TRILL network, a LAN link MAY have multiple access points to this network. All the access RBridges are able to act as the packet forwarder of the VLANs carried on this LAN link. The DRB of this LAN link is responsible to pick out one of the RBridge attached to this LAN link as the appointed forwarder for each VLAN-x. In other words, the DRB assigns VLAN-x to one of the RBridge. For an assigned VLAN, its forwarder is not only responsible for forwarding the packets but also need to store the active MAC addresses of the hosts on this VLAN.

If the VLANs on the LAN link are not appointed properly, some of the RBridges’s MAC tables are easily to be used up while the other RBridges are left idle. Take Figure 2.1 as an example, there are four VLANs carried on the LAN link: w, x, y and z. There are two hosts in both VLAN-w and VLAN-x and one host in both VLAN-y and VLAN-z. RB1 and RB2 are both attached to this LAN link. RB1 is elected as the Designated RBridge who is responsible to choose the appointed forwarder for the above VLANs. The figure shows that VLAN-w,x are assigned to RB1 and VLAN-y,z are assigned to RB2. Obviously, this assignment is not balanced, since the MAC table of RB1 has four entries while the MAC table of RB2 only has two entries. If the DRB can reassign VLAN-w to RB1 and reassign VLAN-y to RB2, both RBridges will have three MAC entries, therefore a more balanced assignment is achieved.

In order to assign the VLANs in a balanced way, the DRB need to know the usage of the MAC tables of its appointed forwarders and the sum of the MAC addresses in each VLAN. Since the RBridges only store the active MAC addresses and a virtual machine can move from one location to another, the MAC entries a VLAN occupy on an RBridge varies from time to time. The assignment of the VLANs cannot be done once for all. It is necessary for the DRB to do the assignment adaptively taking the usage of MAC tables of its appointed forwarders into consideration. Therefore, in Section 5.1, the MAC Entries Report sub-TLV is defined to deliver this kind of information from a forwarder to a DRB.

```
          MAC Entries
          +-----+
          |  w  |
          +-----+
          |  w  |                              MAC Entries
          +-----+ >---+                        +-----+
          |  x  |     |                        |  y  |
          +-----+     |                  +---< +-----+
          |  x  |     |                  |     |  z  |
          +-----+     |                  |     +-----+
                      |                  |
          DRB&AF:w,x  |                  |     AF:y,z
          +-----+     |                  |     +-----+
          | RB2 |-----+                  +-----| RB1 |
          +-----+                              +-----+
             |                                    |
             @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
            @                                          @
           @ +-------+ +-------+   +-------+ +-------+ @
          @  |[H] [H]| |[H] [H]|   |  [H]  | |  [H]  |  @
          @  +-------+ +-------+   +-------+ +-------+  @
           @   VLAN-w    VLAN-x      VLAN-y    VLAN-z  @
            @                                          @
             @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
                                LAN link
```

Figure 2.1: Unbalanced VLAN Assignment

4 Traffic Load Balancing

The traffic from the TRILL network to the local LAN link is called egress traffic while the traffic from the local LAN link to the TRILL network is called ingress traffic. A forwarder RBridge acts as both the ingress and egress point of a VLAN’s traffic. The assignment of the appointed forwarder for each VLAN affects both the egress and ingress traffic load distribution.

4.1 Egress Traffic

One RBridge MAY have multiple ports attached to the same local LAN link. These ports are called "port group" [TRILLbase]. When a DRB assigns a VLAN to an RBridge, its total available egress bandwidth of the port group needs to be taken into consideration. Using the TLV defined in Section 5.2, the load of the egress points are reported from the appointed forwarders to the DRB on the LAN link. The assignment SHOULD NOT cause congestion to an already busy egress point.

After VLAN-x has been assigned to an RBridge, the forwarding port assignment of one of the port group to VLAN-x as the forwarding port is entirely a local matter. Since a LAN link is a STP domain, more than one forwarding port for one VLAN will cause a loop. The forwarder MUST assign one and only one port for each VLAN. Load balancing can be realized through splitting the load among different VLANs as suggested in Section 4.4.4 of [TRILLbase].

4.2 Ingress Traffic

After the known unicast packets enter the TRILL network from the ingress RBridge, they can be sent through the paths starting at this ingress point. Since the DRB knows the whole topology of the TRILL network, it can figure out these paths as well. Therefore, the DRB should take the available bandwidth of these paths into consideration when assigning the appointed forwarder of a VLAN. Any assignment that is possible to congest an already busy ingress point or a path should be avoided.

Traffic Matrices are usually taken as the input to the traffic engineering methods [TE]. The work in this section is actually changing the Traffic Matrices of the TRILL network. If traffic engineering is used in TRILL networks, the forwarder appointment mechanism should work together with the traffic engineering method to in order to achieve a more balanced global traffic distribution of the whole network. The DRB can also collect the probing messages used in the traffic engineering and then assign the VLAN according to the bandwidth utilization. However, the design of this kind of cooperative mechanism for balancing the ingress traffic is left as future work when traffic engineering solutions are begin to be used on TRILL networks [TBD].

5 Definition of sub-TLVs

The Appointed Forwarders TLV has already been defined in [TRILLtlv]. With this TLV, the DRB can appoint an RBridge on the local link to be the forwarder for each VLAN. However, there is no feedback from the appointed forwarder whether the assignment is reasonable. Two sub-TLVs are defined in this section to open the feedback passageway. They can be used by the appointed forwarder to report the number of MAC addresses and traffic load of VLANs in the reverse direction to the DRB. Through the collection of these report messages (these messages can be stored in the MIB of DRB [TRILLmib]), the DRB will have a vision of the MAC tables usage and bandwidth utilization of the RBridges on the LAN link. Based on this vision, the DRB can have a adaptive VLAN assignment.

5.1 MAC Entries Report sub-TLV

The appointed forwarder use MAC Entries Report sub-TLV to report the usage of its MAC table to the DRB. It has the following format:

```
+-+-+-+-+-+-+-+-+
|Type=MACEtrRep |                              (1 byte)
+-+-+-+-+-+-+-+-+
|  Length       |                              (1 byte)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  DRB Nickname                 |              (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Maximum MAC Entries          |              (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Available MAC Entries        |              (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          MAC Entries of VLAN (1)             | (4 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          ......                              | (4 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          MAC Entries of VLAN (N)             | (4 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where each MAC Entries of VLAN is of the form:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RESV  |   VLAN ID             |              (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| The Number of MAC Entries     |              (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

o Type: MAC Entries Report sub-TLV.

o Length: 6+4n bytes, where n is the number of VLANs that the appointed forwarder selects to report their numbers of MAC entries in its MAC table.

o DRB Nickname: The nickname of the Designated RBridge of the local link.

o Maximum MAC Entries: The maximum number of the entries of the MAC table of the appointed forwarder.

o Available MAC Entries: The number of available entries of the MAC table of the appointed forwarder.

o RESV: 4 bits that MUST be sent as zero and ignored on receipt.

o VLAN ID: This field identifies one of the VLANs that assigned to the appointed forwarder.

o The Number of MAC Entries: The number of MAC Entries that the given VLAN occupies in the MAC table of the appointed forwarder. These MAC entries does not only contain the local MACs of the hosts on the local link but also includes the MAC addresses from the same VLAN on the remote link (i.e., the same virtual link).

All the appointed forwarders will report this sub-TLV messages to the DRB of a LAN link. The information contained in these sub-TLV messages will help the DRB to make more balanced VLAN assignment among the RBridges on the LAN link. Because of host mobility, a former balanced VLAN assignment MAY become unbalanced. If a forwarder's MAC table is running out of use, the DRB can remove some VLANs from it and reassign them to another RBridge as the new forwarder. The number of "MAC Entries of VLANs" SHOULD be constrained by the inter-RBridge link MTU that defaults to 1470 bytes. If the MTU is not big enough to hold all the "MAC Entries of VLANs", the appointed forwarder MAY define its own policy to choose which VLANs it wants the DRB to remove [TBD].

5.2 Traffic Bit Rate Report sub-TLV

The appointed forwarder use Traffic Bit Rate Report sub-TLV to report the bandwidth utilization of its port group to the DRB. This sub-TLV has the following format:

```
+-+-+-+-+-+-+-+-+
|Type=TrafficRep|                               (1 byte)
+-+-+-+-+-+-+-+-+
|  Length       |                               (1 byte)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  DRB Nickname                 |               (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Maximum Link Bandwidth       |               (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Available Link Bandwidth     |               (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Traffic Bit Rate of VLAN (1)                 | (4 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ......                           | (4 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Traffic Bit Rate of VLAN (n)                 | (4 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where each Load of VLAN is of the form:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RESV  |   VLAN ID               |              (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Traffic Bit Rate             |              (2 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- o Type: Traffic Bit Rate Report sub-TLV.
- o Length: 6+4n bytes, where n is the number of VLANs that the appointed forwarder selects to report their traffic load that egress onto the port group.
- o DRB Nickname: The nickname of the Designated RBridge of the local link.
- o Maximum Link Bandwidth: The maximum bandwidth of the port group attached to the local link.
- o Available Link Bandwidth: The available bandwidth of the port group attached to the local link.
- o RESV: 4 bits that MUST be sent as zero and ignored on receipt.
- o VLAN ID: This field identifies one of the VLANs that assigned to the appointed forwarder.
- o Traffic Bit Rate: The traffic bit rate of the given VLAN onto the local link through the port group of the appointed forwarder.

The appointed forwarder send messages of this sub-TLV to its DRB. The DRB will know the bandwidth utilization of the port group of the appointed forwarder. If the port group of an RBridge attached to the local link is already heavily used, the DRB will refrain from assigning additional VLANs to this RBridge. If an appointed forwarder’s port group attached to the local link is congested, its DRB MAY remove some of the VLANs reported in the Traffic Bit Rate Report TLV message and reassign these VLANs to other RBridges attached to the same local link, which will decrease the traffic bit rate via that RBridge. The policy to decide which VLANs to reassign is [TBD].

6 Security Considerations

The delivery of the messages types in this document can be protected with the cryptographic mechanism proposed in [RFC5310]. In the future, TRILL MAY define its own secure control message transmission. The new message types introduced in this document can make use of that secure channel.

7 IANA Considerations

Two code points of IS-IS sub-TLVs need to be assigned. This work should be done in conjunction with the work of [TRILLtlv].

8 References

8.1 Normative References

[TRILLbase] R. Perlman, D. Eastlake, D.G. Dutt, S. Gai and A. Ghanwani, "RBridges: Base Protocol Specification", draft-ietf-trill-rbridge-protocol-16.txt, working in progress.

[TRILLtlv] D. Eastlake, A. Banerjee, D. Dutt, R. Perlman and A. Ghanwani, "TRILL Use of IS-IS", draft-ietf-trill-adj-02.txt, working in progress.

[TRILLmib] A. Rijhsinghani, K. Zebrose, "Definitions of Managed Objects for RBridges", draft-ietf-trill-rbridge-mib-02.txt, working in progress.

[RFC5310] M. Bhatia, V. Manral, T. Li, et at., "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

8.2 Informative References

[CAMtable] B. Hedlund, "Evolving Data Center Switching", http://internetworkexpert.s3.amazonaws.com/2010/trill1/TRILL-intro-part1.pdf

[SAN] "Configuring an iSCSI Storage Area Network Using Brocade FCX Switches", Brocade CONFIGURATION GUIDE, 2010.

[VL2] A. Greenberg,J.R. Hamilton,N Jain, et al., "VL2: A scalable and flexible data center network", in Proceedings of ACM SIGCOMM, 2009.

[TE] M. Roughan, M. Throup, and Y. Zhang, "Traffic Engineering with Estimated Traffic Matrices" , in Proceedings of ACM IMC, 2003.

Author's Addresses

Mingui Zhang
Huawei Technologies Co.,Ltd
HuaWei Building, No.3 Xinxi Rd., Shang-Di
Information Industry Base, Hai-Dian District,
Beijing, 100085 P.R. China

Email: mingui@huawei.com

Dacheng Zhang
Huawei Technologies Co.,Ltd
HuaWei Building, No.3 Xinxi Rd., Shang-Di
Information Industry Base, Hai-Dian District,
Beijing, 100085 P.R. China

Email: zhangdacheng@huawei.com