

Network Working Group
INTERNET-DRAFT
Intended status: Informational
Expires: 8 June 2009

Donald Eastlake 3rd
Stellar Switches
9 December 2008

Rbridge Notes

<[draft-eastlake-trill-rbridge-notes-01.txt](#)>

Status of This Document

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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

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

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

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

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

Abstract

This document provides additional informational material related to RBridges, which are devices that implement the TRILL protocol. It is a supplement to the RBridges base protocol specification and includes discussion of tradeoffs in some features and configurations of RBridges. In addition, it provides a sketch of a proof that, with reasonable assumptions, persistent loops do not occur in a TRILL campus.

INTERNET-DRAFT

TRILL Header Options

Table of Contents

Status of This Document.....	1
Abstract.....	1
Table of Contents.....	2
1 . Introduction.....	3
2 . Zero Configuration Comparison.....	4
2.1 IEEE 802.1D Bridges.....	4
2.2 IEEE 802.1Q Bridges.....	4
2.3 RBridges.....	6
3 . Pluses and Minuses.....	8
3.1 Trees and the Forest.....	8
3.2 Loop Safety.....	9
3.2.1 Loop Safety Mechanisms.....	9
3.2.2 Loop Safety Tradeoffs.....	10
4 . No Persistent Loops.....	11
4.1 Categories of Loops.....	11
4.2 Analysis for a Single Pair of RBridges.....	12
4.3 Analysis for on Arbitrary Bridged LAN.....	14
5 . Security Considerations.....	18
6 . Normative References.....	18
7 . Informative References.....	18
8 . IANA Considerations.....	18
Disclaimer.....	19
Additional IPR Provisions.....	19
Author's Address.....	20
Expiration and File Name.....	20

1. Introduction

This document provides informational material related to RBridges, which are devices that implement the TRILL protocol.

[Section 2](#) briefly compares some aspects of zero configuration IEEE 802.1D and 802.1Q bridges and RBridges. [Section 3](#) discusses some tradeoffs in features and configurations of RBridges. While [Section 4](#) presents a sketch of a proof that, with reasonable assumptions, persistent loops do not occur in a TRILL campus.

The terms and acronyms defined in Sections [1.3](#) and [1.4](#) of [\[RFCprotocol\]](#) are used with the same definitions herein.

[2.](#) Zero Configuration Comparison

This section provides an informational comparison of the behavior of a zero configuration IEEE 802.1D bridge, a zero configuration IEEE 802.1Q bridge, or a zero configuration RBridge, in a network of possibly configured devices. The goal is to clarify the behavioral differences, particularly in regard to VLANs and priority. All three devices can learn end station addresses in essentially the same way, through the observation of traffic (although RBridges provide an additional facility, which can be configured to learn such information via ESADI messages).

[2.1](#) IEEE 802.1D Bridges

802.1D bridges [[802.1D](#)] are ignorant of VLANs. They are unaware whether frames received have a C-tag (formerly called Q-tag), the tag which provides VLAN ID and priority. As a result, 802.1D bridges learn end station address locations based on simple 48-bit MAC addresses unqualified by VLAN. (Actually 47 significant bits as addresses with the group bit on are not learned.)

In a bridged LAN with 802.1D bridges, a single spanning tree is determined and frames must flow along this tree. As a result, only links that are part of the spanning tree can be used for through traffic.

Since 802.1D bridges are ignorant of priority, frames do not get re-ordered based on priority, low priority frames do not get preferentially discarded due to the favoring of high priority frames, and there are no facilities for mapping priority levels.

[2.2](#) IEEE 802.1Q Bridges

802.1Q bridges [[802.1Q](#)] are aware of VLANs. Every frame internal to such a bridge has a VLAN ID and priority associated with it. If the frame arrived without a VLAN tag, the bridge port logic either drops the frame or associates a VLAN and priority with it (see [[RFCprotocol](#)] [Appendix D](#)). When a frame is transmitted by an 802.1Q bridge, it can be sent with a VLAN tag indicating its VLAN and priority or without such a tag, depending on port configuration. Frames are only transmitted through a port if the VLAN of the frame is in the set of VLANs enabled on that port. For a zero configuration port, that set consists of only VLAN 1.

The learning of end station addresses in such a bridge is based on a combined 12-bit VLAN ID and 48-bit (actually 47-bit as above) MAC

address. (However, 802.1Q bridges may support mapping VLAN IDs into a smaller number of learning table IDs so that learning between those VLANs is shared (see Section 4.6.3 of [[RFCprotocol](#)].))

A zero configuration 802.1Q bridge accepts frames that arrive tagged with any valid VLAN. (They drop frames tagged with the illegal VLAN 0xFFFF.) Frames without a VLAN tag are associated with VLAN 1. However, because the ports of a zero configuration bridge have only VLAN 1 enabled for output, accepted frames for any VLANs other than 1, unless they are addressed to the bridge itself, have no place they can go. As a result, in an 802.1Q bridge network where there is bridged traffic in any VLAN other than the default VLAN 1, it is essential to configure ports to permit output for these other VLANs. This can be done through management at each bridge or with VLAN Registration Protocol messages (GVRP [[802.1Q](#)] or MVRP [[802.1ak](#)]) or a

combination of these techniques (see Section 4.7.2 of [[RFCprotocol](#)]).

By default, 802.1Q bridges form a single spanning tree and frames flow along that tree. The bridge ports on spanning tree inter-bridge links must be configured to enable all the VLANs which require the link for connectivity. 802.1Q bridged LANs can be configured to have up to an additional 64 spanning trees with traffic segregated between the trees based on VLAN; however, the above considerations would apply within each of such multiple spanning trees.

The recommended default for 802.1Q bridge ports is that VLANs be disabled by default but dynamically registerable, except for VLAN 1, which is fixed registered. As a result, VLAN Registration Protocol frames can generally flow along the spanning tree adding the needed VLANs to the ports where they are received so as to provide connectivity between all end stations in each VLAN.

Since they recognize priority, 802.1Q bridges can re-order frames to expedite those of higher priority and discard lower priority frames in preference to discarding higher priority frames. 802.1Q bridge ports not only associate a priority code point (0 through 7, default 1) with any frame received without a C-tag, they also map the priority of a frame received with a C-tag. By default, this mapping (which in IEEE 802.1 is called "regeneration") is the identity mapping but can map each received priority code point to an arbitrary other frame priority code point.

This priority mapping would make it possible, for example, to configure a bridged LAN so that it had regions in which priority code points had different external semantics such that the priority associated with a frame was mapped at the regions boundaries. This would require careful configuration of the appropriate ports of all bridges at inter-regional boundaries.

[2.3](#) RBridges

RBridges are VLAN tag aware and, in terms of VLANs and priority, the port behavior and configurability of an RBridge port is identical that of an 802.1Q customer bridge except for the handling of VLAN registration protocols (GVRP and MVRP). RBridges also learn end station addresses based on a combined 12-bit VLAN ID and 48-bit MAC

address (actually 47-bit as above).

Zero configuration RBridges accept TRILL frames for any valid VLAN but accept native frames only on a port where they are appointed forwarder for the frame's VLAN. Native frames are not forwarded in native form out of any local port unless the RBridge is the appointed forwarder for the port and VLAN. In the zero configuration case, it could only be appointed forwarder for VLAN 1. However, once a native frame is encapsulated into a TRILL data frame, it is not restricted to ports where output to its Inner.VLAN is enabled.

A zero configuration RBridge could forward TRILL data frames and encapsulate and forward native frames to another RBridge or RBridges. For example, a known unicast TRILL data frame would be forwarded toward the correct egress RBridge even if it is in a VLAN other than 1. An RBridge would distribute a multdestination frame on its distribution tree, possibly pruned for efficiency, to a subset of RBridges. This is because an RBridge to RBridge link does not need its end ports to have the relevant end-to-end VLANs added to them; the encapsulation can tag the frame with the Designated VLAN as the outer VLAN ID.

As a result of this, there is normally no need for VLAN Registration Protocol frames to affect the receiving ports of transit or egress RBridges. It can, however, still be useful for such frames to add VLANs to the set for the ports of ingress RBridges where they are received. Also, it can be useful for RBridges to send such VLAN Registration Protocol frames to bridges (or possibly even end stations) that may be included in the campus. (See Section 4.7.2 of [\[RFCprotocol\]](#) for more details on RBridge handling of dynamic VLAN registration.)

In terms of frame priority, RBridges associate a priority code point with every native frame they receive in the same way that an IEEE 802.1Q bridge does. They assign a priority if the frame is untagged. If the frame is tagged and thus has a priority code point, they map it to a potentially different priority code point, although the default mapping is the identity mapping. While this determines the priority of a native frame, if the frame received is a TRILL data or ESADI frame, it contains an Inner.VLAN tag with the priority of the frame at the time it was TRILL encapsulated. This inner priority code point is used in the case of TRILL data and ESADI frames. (Priority is not relevant for a core TRILL IS-IS frame received by an RBridge.)

This use of the Inner.VLAN priority code point for forwarded TRILL frames means that, in some sense, the interpretation of priority code points should be uniform throughout a campus.

[3. Pluses and Minuses](#)

The subsections below examine the tradeoffs in various RBridge features and configuration options.

[3.1 Trees and the Forest](#)

Although one distribution tree is logically sufficient to distribute multi-destination frames in a campus, TRILL supports multiple distribution trees for the following reasons:

1. It is desirable to allow choosing a different distribution tree than the one rooted at the ingress RBridge for some frames in order to allow multipathing of multi-destination traffic encapsulated by a particular RBridge. (See [\[RFCprotocol\]](#) [Appendix C.](#))
2. Using a tree rooted at the ingress RBridge optimizes the distribution path and (almost always) the cost of delivery when the number of destination links is a subset of the total number of links, as is the case with VLANs and IP multicasts.
3. For unknown unicast destinations, using a tree rooted at the ingress RBridge minimizes out-of-order delivery because, in the case where a flow starts before the location of the destination is known by the RBridges, the path to the destination is the same as the shortest path to the destination (unless equal cost multipath is being used).

A distribution tree rooted in the ingress RBridge is not always the best choice. To assure availability of such a tree, it would be necessary to compute a tree rooted at every RBridges. But a different tradeoff might be wanted in terms of the expense of computing many trees versus optimality of traffic distribution, so fewer trees would be desired.

As described in [\[RFCprotocol\]](#) [Section 4.3](#), each RBridge includes in its LSP a priority for itself to be chosen as a distribution tree root and a number of distribution trees. Ties in priority are broken by System ID. The number of trees specified by the RBridge that is

highest priority (lowest numeric priority / system ID) to be a distribution tree root governs the campus. RBridge computes the specified number, say n, trees rooted at the n RBridges that are highest priority to be a tree root. In a zero configuration RBridge campus, each RBridge calculates two trees, one rooted at each of the two RBridges with lowest System IDs, and each RBridges distributes multi-destination frames which it ingresses over the tree whose root is least cost from the ingress RBridge.

[3.2](#) Loop Safety

Avoidance of loops at Layer 2 is critical as they lead to rapid network saturation, denial of service, and even exponential growth in the number of frames looping.

The asynchronous and distributed nature of the processes in RBridges and bridges and the imperfections of these devices and communications paths between them make absolute guarantees of delivery, frame ordering, or transient loop avoidance impossible. However, the default loop safety provisions of TRILL, under the assumptions TRILL makes, are intended to provide the same order of reliability in loop avoidance as modern bridged LANs.

[3.2.1](#) Loop Safety Mechanisms

There are two primary safety mechanisms used by RBridges to protect against persistent loops. (There are also additional mechanisms to greatly reduce the occurrence and severity of transient loops.) The primary persistent loop safety mechanisms are as follows:

- o The use of TRILL IS-IS Hellos, and
- o The decapsulation check.

The adequacy of the default set of TRILL Hellos to protect against persistent loops is discussed in [Section 4](#) below.

The second mechanism is the optional "decapsulation check" (sometimes called the "root bridge collision" check). Every RBridge is required to report in its link state for each VLAN for which it is appointed forwarder on at least one of its ports, the complete list of root bridges it sees on those ports. (This list may be null if none of

those ports leads to a bridged LAN.)

When an egress RBridge is about to decapsulate a TRILL data frame and send a VLAN-x native frame out a port and it sees a root bridge R out that port, it may optionally check to see if that R is on the list of root bridges seen for VLAN-x by the frame's ingress RBridge. If this check finds R, then the checking RBridge knows that it was about to decapsulate onto either (1) the same bridged LAN from which the native frame originated, possibly forming a loop, or (2) onto a bridged LAN that was also directly connected to the ingress RBridge on a port where the ingress RBridge was appointed forwarder for the frame's VLAN. In this second case, the ingress RBridge should have already forwarded the frame locally and so it should not have arrived at the egress RBridge in encapsulated form. In any case, if this optional check is performed and the locally observed root bridge is found in the ingress RBridge's list for the frame's VLAN, the egress

Rbridge does not send the decapsulated native frame out the port but discards it.

[3.2.2](#) Loop Safety Tradeoffs

The transmission and reception of many TRILL IS-IS Hellos can impact available communications bandwidth and processing power. In other words, they can stress the control plane. On the other hand, use of the decapsulation (root bridge collision) check requires an additional check in the data plane before any TRILL data is decapsulated onto a link, making the data plane more complex.

If the computational and bandwidth load are acceptable, a campus will be safer to the extent the RBridges are configured to perform the decapsulation check and also send Hellos on at least the default set of VLANs as specified in [[RFCprotocol](#)] [Section 4.2.3.1](#).

Under normal circumstances, if any of the RBridges connected to a link are configured to send Hellos into the link on fewer than the default set of VLANs, it is recommended that those RBridges implement and use the decapsulation check on their ports connected to that link.

Under special circumstances, where it is known to be safe with a high

degree of reliability, RBridges may be configured to send Hellos on fewer VLANs than the default without using the decapsulation check.

[4.](#) No Persistent Loops

This section demonstrates that, with reasonable assumptions, the default set of Hellos that RBridges send do not permit the occurrence of persistent loops in an RBridge campus.

[Section 4.1](#) below divides cases where a frame persistently loops into three categories and show that only one of these can be problematic. [Section 4.2](#) discusses the possibly problematic case from the point of view of a single pair of connected RBridges and provides a sketch of a proof that, for any pair of connected RBridges and under reasonable assumptions, the problematic case cannot persist. Finally [Section 4.3](#) expands this sketch of a proof to an arbitrary bridged LAN connected to an arbitrary number of RBridges.

[4.1](#) Categories of Loops

An RBridge campus can consist of a large number of RBridges (in principle somewhat less than 2^{16} or more if some do not require nicknames) interconnected by LANs that may be bridged. The RBridge ports and any bridges involved could be arbitrarily configured concerning what VLANs they pass, how they treat untagged frames on frame receipt and for what VLANs they strip VLAN tags on transmission.

A persistent loop would be a frame that cycled indefinitely, although it might, at various parts of that cycle, be tagged with different VLANs and might be in TRILL encapsulated form or native form. Persistent loops can be divided into three categories as follows:

1. The first category of persistent loop would be one within a bridged or unbridged LAN between RBridges or end stations. The looping frame could be any type of frame, native, TRILL, or control. This category is the concern of IEEE 802.1 bridging standards. They solve this potential problem by forwarding frames, when they are forwarded, in accordance with one of several variations of the spanning tree protocol. While transient loops can occur due to loss of spanning tree BPDUs or topology changes that are not immediately detected, spanning tree prevents persistent loops unless unsafe bridge options, such as inhibiting the transmission of BPDUs, are used.
2. The second category of persistent loop would be one of TRILL frames persistently transiting the same set of at least two RBridges. The frames cannot loop in the network between RBridges as that would be a category one loop discussed above. Also, there is no way for core TRILL IS-IS frames to loop as they only go one RBridge hop and are never forwarded by an RBridge. So such a loop

would have to be of a TRILL data or TRILL ESADI frame among RBridges. Such persistent loops cannot occur because TRILL uses IS-IS, which does not produce persistent loops in the forwarding of unicast frames or in the trees constructed for the distribution of multi-destination frames.

In addition, all TRILL data and ESADI frames have a TTL that must be decremented by at least one each RBridge hop and the frame discarded, rather than forwarded, if the TTL is reduced to zero. Therefore no individual frame can persistently loop.

Although not necessary to avoid persistent loops as herein defined, RBridges further inhibit possible temporary looping of multi-destination TRILL frames through the adjacency checks, including the reverse path forwarding check, made on arriving TRILL data or ESADI frames.

3. There remains only one further category for persistent loops. In category 1 above, we discuss why there cannot be persistent loops within the possibly bridged LANs which connect RBridges. Therefore the loop must involve frames sent between RBridges. In category 2 above, we discuss why there cannot be persistent loops of TRILL frames being transmitted between RBridges. Therefore, any persistent loop must involve, at least in part, non-TRILL frames transmitted between RBridges.

There are only two non-TRILL types of frames, control frames and native frames. Control frames are transmitted only one RBridge or bridge hop and are not forwarded so they cannot loop. Therefore, any persistent loop must involve a native frame sent from one RBridge to another RBridge. Note that native frames do not have TTL protection.

[Section 4.2](#) below gives a sketch of a proof that native frame type 3 persistent loops cannot occur by considering a single pair of RBridges on a link. [Section 4.3](#) goes into excruciating detail extending this to an arbitrary set RBridges connected to an arbitrary bridged LAN.

[4.2](#) Analysis for a Single Pair of RBridges

It is shown above that any persistent loop in an RBridge campus must involve a native frame sent from one RBridge to another. These must be different RBridges as it is a fundamental assumption of the Ethernet service model that a frame transmitted on an Ethernet link will not be received by the transmitter.

For there to be a loop, the receiving RBridge must actually accept

the frame and forward it in some form. An RBridge only accepts a native frame if it is appointed forwarder on the port for the frame's

VLAN. If it is not the appointed VLAN-x forwarder for the native frame, the native frame is simply discarded.

An RBridge never transmits a native frame unless it is appointed forwarder for the frame's VLAN on the port where the frame is transmitted.

Thus, for there to be a loop involving native frame transmission between RBridges, both must be appointed forwarder on the link. However, they would not necessarily have to be appointed for the same VLAN. For example, the transmitting RBridge or some bridge along the way could be stripping VLAN tags and some later bridge or the receiving RBridge could insert a different VLAN tag or associate the frame with a different VLAN.

Can this situation occur?

The primary defenses against such dual appointed forwarder situations are, as described in [\[RFCprotocol\] Section 4.2.3.1](#), the DRB and its appointer forwarder determinations, which are mediated by TRILL IS-IS Hellos. By default, the ports on which Hellos are transmitted include any port where an RBridge is an appointed forwarder. Hellos are sent out such a port for each VLAN for which the RBridge is appointed forwarder.

Assume the RBridge sending the native frame is RB-s and the RBridge receiving it is RB-r. Since a native frame is getting from RB-s to RB-r, we assume that a Hello sent in the same VLAN will also get from RB-s to RB-r and arrive with the same VLAN as the native frame. (This might not be true if successive bridge/RBridge ports were configured so that the Outer.VLAN was stripped and then frames were assigned a VLAN based on frame protocol with different VLANs for TRILL IS-IS frames (VLAN-T) and for a possibly looping native frame (VLAN-n). Such a situation would not necessarily cause a loop but could if other conditions were met including that no VLAN-n Hellos were being sent from RBridges and received by RB-r. In any case, we assume that this is not the situation.)

It will be clear to RB-r from the Hello it receives from RB-s that RB-s considers itself to be the appointed forwarder as there is a flag in the Hello for this purpose. If the Hello is received with a different Outer.VLAN ID from its Inner.VLAN ID, then VLAN mapping is occurring and, as stated in [\[RFCprotocol\]](#), native frames received by RB-r in VLAN-n will be discarded. Thus there can be no loop with VLAN mapping.

Without VLAN mapping, VLAN-T equals VLAN-n and RB-r will receive Hellos from RB-s indicating that RB-s believes itself to be appointed

forwarder for the VLAN. This will cause RB-s to inhibit its appointed forwarder activity and discard the native frame, so no loop is formed.

[4.3](#) Analysis for on Arbitrary Bridged LAN

When a link provides full bi-directional connectivity between the RBridges connected to it, each such RBridge can see the Hellos sent by the others on that link. In that case, the selection of the one DRB on the link and the decisions by that DRB as to appointed forwarders are straightforward. However, the exact situation on an arbitrary bridged LAN connecting multiple RBridges can, in the worst case, be much more complex than this.

Of course, with N RBridges attached to a bridged LAN, the analysis in [Section 4.2](#) applies to each $N*(N-1)/2$ unordered pairs. Thus, it is clear from the outset that there cannot be any loops. Nevertheless, it is interesting to analyze the different populations of RBridges there can be attached to the bridged LAN link.

We will model the transmission of Hellos between the N RBridges connected by an arbitrary bridged LAN as the existence of a pipe between each pair of RBridges. Thus there are $N*(N-1)/2$ such pipes. When an RBridge sends a Hello, it is pushed into all the pipes terminating at that RBridge. Each pipe passes frames tagged with an arbitrary subset of legal VLAN IDs. In general, the set of VLANs passed can be asymmetric, that is, it is different in each direction through the pipe.

TRILL IS-IS Hellos have the ID of the VLAN on which they are sent embedded in the body of the Hello. In this section, we will initially assume that no VLAN mapping is occurring. With this assumption, we need not worry about Outer.VLAN tags getting stripped or added by ports. By the time a TRILL IS-IS Hello arrives at RBridge specific code as shown in [\[RFCprotocol\]](#) Figure 4.3, it will have had a VLAN ID associated with it. The no-VLAN-mapping assumption implies that this will necessarily be the VLAN ID with which it was transmitted.

Consider the set of N RBridges connected to a bridged LAN: RB1, RB2, disabled or have no VLANs enabled do not count as a connection to the link to which they are physically attached.) These RBridges can be strictly ordered by their priority to become DRB. This priority is an unsigned 55-bit integer consisting of the RBridge's 7-bit priority to become DRB (the IS-IS Hello header DIS priority) concatenated with

its 48-bit port MAC address. (There actually should be only 54 significant bits as the group bit in the MAC address should always be zero.) Assume, without loss of generality, that the RBridges are numbered in priority order so that RB1 is the highest priority to

become DRB.

RB1 will specify a Designated VLAN in its Hellos and will specify the adjacencies that it sees on its Designated VLAN in the Hellos that it sends on that VLAN. The set of RBridges receiving Hellos from RB1 on any VLAN will be denoted {H+RB1}. The RBridges in {H+RB1} will see that RB1 is DRB and will defer to RB1 since, by construction, they are of lower priority. If they have the Designated VLAN enabled on the port on which they receive a Hello from RB1, they will send such a Hello on the Designated VLAN on that port. If that Hellos makes it through to RB1, the normal IS-IS mechanisms will establish IS-IS adjacencies between them and RB1 and transitively among this subset of the RBridges with connectivity to RB1 over its Designated VLAN. Thus they will be able to exchange TRILL data, LSPs, etc. In the normal case, where the bridged LAN conforms to the assumptions in [\[RFCprotocol\] Section 2.3](#), this will be all of the RBridges connected by this bridged LAN. It is only in the case of badly configured bridges, that is, configurations which violate the the assumptions in [Section 2.3](#), that the discussion below in this section is relevant.

There may be other RBridges in {H+RB1} that can see Hellos from RB1 on one or more VLANs but cannot establish an IS-IS adjacency with it on the Designated VLAN as above and are orphaned from the link on that port. This can occur for two reasons: (1) because the Designated VLAN is not enabled on their RBridge port connected to the link, or (2) because, although the Designated VLAN is enabled, there is only connectivity for the VLAN through the bridged LAN from RB1 to the RBridge in question but no connectivity in the other direction.

There may be RBridges connected to the bridged LAN which cannot see Hellos from RB1. That is, in {RB*} but not in {H+RB1}. Such RBridges have no knowledge of RB1 and thus could not defer to it as DRB. However, they may receive a Hello from a member of {H+RB1} other than RB1; that is, in effect, they may be two "Hello ops" from RB1. If they receive such Hello from an RBridge that is higher priority than they are, they will defer and know that they are not DRB. They do not have connectivity from RB1 over the Designated VLAN because, if they did, they would have received Hellos from RB1 and be in {H+RB1}. Thus

they are orphaned from the link. Denoting such RBridges as $\{H^{++}RB1\}$, there may be yet further RBridges in $\{RB^*\}$ that are not in $\{H^{++}RB1\}$ or $\{H^{++}RB1\}$ but which can receive Hellos from a higher priority RBridge in $\{H^{++}RB1\}$ and which they will defer to. Such RBridges are three Hello hops from RB1, are similarly orphaned, are denoted as $\{H^{+++}RB1\}$. This process may continue if there are further RBridges even more Hello hops from RB1 such that there is a chain of RBridges from them to RB1 with increasing priority as RB1 is approached. We will denote the union of RB1, $\{H^{++}RB1\}$, $\{H^{+++}RB1\}$, etc., will be denoted as $\{H^*RB1\}$, i.e. the set of all RBridges on the link which either are RB1 or which defer to RB1 as DRB or are at the end of a chain of RBridges each of which defers to the next ending in RB1.

In the worst case, there could be configuration such that RB1 is DRB but cannot form an adjacency with any other RBridge, RB2 will defer to RB1 because it sees Hellos from RB1, RB3 defers to RB2 because it sees Hellos from RB2 but not RB1, RB4 defers to RB3 because it sees Hellos from RB3 but not RB1 or RB2, etc., through RBN which defers to RB(N-1) because it sees Hellos from RB(N-1) but not from any lower numbered RBridge.

Consider the RBridges in $\{RB^*\}$ but not in $\{H^*RB1\}$. If this set is not empty, there will be one RBridge in this set that is highest priority to be DRB, say RBi. Since, by construction, RBi is not in $\{H^*RB1\}$, it is not receiving Hellos from any higher priority RBridge and thinks itself to be DRB. We can now perform the same analysis above leading to the conclusion that there will be a set (possibly consisting of just RBi) of RBridges $\{H^*RBi\}$ which (1) receive Hellos from RBi or a deferral chain ending in RBi via one or more Hello hops, (2) do not receive Hellos from RB1 or from any RBridge with a higher priority in $\{H^*RB1\}$. Thus the members of $\{H^*RBi\}$ directly or indirectly defer to RBi as DRB.

There are several possibilities for connectivity between the $\{H^*RB1\}$ and $\{H^*RBi\}$ sets:

1. It may be that there is no connectivity at all between RBridges in $\{H^*RB1\}$ and $\{H^*RBi\}$ in which case each of these sets is connected to what is, for all practical purposes, a separate link. In this case it is loop safe that there is a separate DRB for each (RB1 and RBi) and there may be a different appointed forwarders in each set for the same VLAN. In this case a native frame sent by an RBridge in either set cannot get to an RBridge in the other set

since, in this case, there is no connectivity.

2. Alternatively, there may be some unidirectional or bi-directional connectivity between one or more RBridges in $\{H*RB1\}$ and $\{H*RBi\}$. However, in no case could there be connectivity from a higher priority RBridge in $\{H*RB1\}$ to a lower priority RBridge in $\{H*RBi\}$ as that would cause the lower priority RBridge to leave $\{H*RBi\}$ and join $\{H*RB1\}$, deferring directly or indirectly to RB1. However, other possible connectivity is still potentially dangerous. In particular, connectivity over VLAN-x between two RBridges that are both VLAN-x appointed forwarders for that VLAN causes a loop unless one of the appointed forwarders is inhibited. There are three possibilities:
 - 2a. Unidirectional connectivity from a lower priority RBridge in the set with the lower priority DRB ($\{H*RBi\}$) to a higher priority RBridge in the set with a higher priority DRB ($\{H*RB1\}$).
 - 2b. Unidirectional connectivity from a lower priority RBridge in the set with the higher priority DRB to a higher priority RBridge in the set with a lower priority DRB.

2c. Bidirectional connectivity as in 2b.

The danger possible in 2a through 2c is solved by providing that while a higher priority RBridge is receiving Hellos on VLAN-x from a lower priority RBridge where the lower priority RBridge is an appointed forwarder for VLAN-x, the higher priority RBridge does not encapsulate native frames off the link or decapsulate native frames onto the link for VLAN-x. This is one reason why all RBridges, by default, send Hellos on all VLANs for which they are appointed forwarder.

Even outside of the union of $\{H*RB1\}$ and $\{H*RBi\}$, there may be additional RBridges connected to the bridged LAN. If so there would be one of them with highest priority, say RBj. We can then repeat the above analysis and see that there is a set $\{H*RBj\}$ of RBridges deferring directly or indirectly to RBj. As above, if there is no connectivity between any RBridge in $\{H*RBj\}$ and any RBridge in either $\{H*RB1\}$ or $\{H*RBi\}$, then these populations of RBridges can act independently without risk of a loop. However, if there is connectivity on VLAN-x from a VLAN-x appointed forwarder in $\{H*RBj\}$ to one in $\{H*RB1\}$ or $\{H*RBi\}$ then the higher priority of the conflicting appointed forwarders inhibit any encapsulation or

decapsulation of any VLAN-x native frames off of or onto the link.

This process can be continued as long as their remain RBridges connected to the bridged LAN in question which are not yet found to be part of a set of RBridges deferring directly or indirectly to a DRB. The method of construction for these sets outlined above means that the sets will be produced in order of declining priority of the set's DRB. By construction, they can be no persistent connectivity, unidirectional or otherwise, from a higher priority RBridge in a set with a higher priority DRB to a lower priority RBridge in a set with a lower priority DRB as it would cause the lower priority RBridge to switch sets. Any of these sets of RBridges can safely act independently if they have no connectivity over the bridged LAN to any RBridges in any other set. Whenever there is connectivity over VLAN-x between two RBridges that are appointed VLAN-x forwarder, the higher priority RBridge of the two does not encapsulate or decapsulating VLAN-x native frames off of or onto the link.

The addition of VLAN mapping, ignored in the analysis above, makes more complex loop possible. For example, if mappings form a cycle, there could be loops in the campus where a frame is decapsulated from VLAN-x, encapsulated as VLAN-y, then decapsulated from VLAN-y and encapsulated as VLAN-x (x->y->x) or longer loops going through more VLANs. However, as specified in [\[RFCprotocol\] Section 4.2.3.1.3](#), when VLAN mapping is detected, the "to" VLAN ID is disabled at the detecting RBridge. Thus, a loop cannot be formed via a VLAN mapping path through a bridged LAN between RBridges because the mapping would inhibit processing of frames in the receiving RBridge.

[5](#). Security Considerations

This document provides additional informational notes related to RBridges and the TRILL protocol but not directly related to security. For RBridge base protocol security considerations, see [\[RFCprotocol\]](#).

[6](#). Normative References

[RFCprotocol] - "Rbridges: Base Protocol Specification", R. Perlman et al, [draft-ietf-trill-rbridge-protocol-10.txt](#), November 2, 2008, work in progress.

[7.](#) Informative References

- [802.1ak] "IEEE Standard for Local and metropolitan area networks / Virtual Bridged Local Area Networks / Multiple Registration Protocol", IEEE Standard 802.1ak-2007, 22 June 2007.
- [802.1D] "IEEE Standard for Local and metropolitan area networks / Media Access Control (MAC) Bridges", IEEE Standard 802.1D-2004, 9 June 2004.
- [802.1Q] "IEEE Standard for Local and metropolitan area networks / Virtual Bridged Local Area Networks", IEEE Standard 802.1Q-2005, 19 May 2006.

[8.](#) IANA Considerations

This document requires no IANA actions.

RFC Editor: This section should be deleted before publication.

Disclaimer

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS

OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Additional IPR Provisions

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

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

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Copyright (C) The IETF Trust 2008 This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Author's Address

Donald E. Eastlake 3rd
Stellar Switches
155 Beaver Street
Milford, MA 01757 USA

email: d3e3e3@gmail.com

Expiration and File Name

This draft expires in 8 June 2009.

Its file name is [draft-eastlake-trill-notes-01.txt](#).

